Design and control of automated guided vehicle systems: A case study

Citation for published version (APA):

Li, Q., Adriaansen, A. C., Udding, J. T., & Pogromski, A. Y. (2011). Design and control of automated guided vehicle systems: A case study. In *Proceedings of the 18th World Congress of the International Federation of Automatic Control (IFAC)*, 28 August - 2 September 2011, Milan, Italy (pp. 13852-13857). IFAC.
<https://doi.org/10.3182/20110828-6-IT-1002.01232>

DOI:

[10.3182/20110828-6-IT-1002.01232](https://doi.org/10.3182/20110828-6-IT-1002.01232)

Document status and date:

Published: 01/01/2011

Document Version:

Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Design and Control of Automated Guided Vehicle Systems: A Case Study

Q. Li, A. C. Adriaansen, J. T. Udding, and A. Y. Pogromsky *

** Department of Mechanical Engineering, Eindhoven University of Technology, 5600 MB, The Netherlands (Tel: +31-40-2473578; e-mail: q.li@tue.nl).*

Abstract: In this paper, we study the design and control of automated guided vehicle (AGV) systems, with the focus on the quayside container transport in an automated container terminal. We first set up an event-driven model for an AGV system in the zone control framework. Then a number of layouts of the road network (guide-path) are carefully designed for the workspace of the AGVs in a container terminal. Based on our zone control model, a traffic control strategy is proposed, which decouples the motion conflict resolution among the AGVs from the routing problem. A routing algorithm is also constructed with the goal of minimizing the vehicle travel distances and the transportation time. The efficiency of the integrated design and control is demonstrated by computer simulations in terms of a set of defined measures of system performance. Lastly, we point out several possibilities towards improving our current results.

1. INTRODUCTION

Automated guided vehicles normally mean mobile robots (or unmanned vehicles) used in transporting objects. They were traditionally employed in manufacturing systems, but have recently extended their popularity to many other industrial applications, such as goods transportation in warehouses and container transshipment at container terminals. See Vis [2006] for an comprehensive survey of the research on the design of AGV systems.

The booming of international trade spurs the development of automated container terminals (ACTs), equipped with automated container transshipment systems (consisting of automated cranes and automated guided vehicles etc.), that can meet rapidly increasing demands for higher operational efficiency, lower costs, and smaller variability than what traditional terminals can achieve. There are four main issues in building an AGV system in an ACT. The first is the design of the guide-path (we call it road network in this paper) that specifies possible paths on which the vehicles can travel [Steenken et al., 2004]. The second is the dispatching problem which is about where and when vehicles should go for the container loading or discharging tasks [Briskorn et al., 2006, Kim and Bae, 1999, Bish et al., 2005, Nguyen and Kim, 2009]. The third is the vehicle routing aimed at finding good paths for vehicles dispatched for certain tasks [Steenke et al., 1993, Duinkerken et al., 2006, Stahlbock and Voß, 2008a]. The last is the conflict resolutions among the fleet of vehicles and between the vehicles and other container handling equipments [Evers and Koppers, 1996, Lehmann et al., 2006, Kim et al., 2006]. Although all these matters are interconnected as far as the system performance is concerned, it is very difficult to take them into a comprehensive consideration. As a meaningful

step towards a complete solution, in this paper, we present a package of designs and controls that address the first, third and fourth issues simultaneously for the quayside container transportation.

Our contributions can be summarized as follows. We first establish a zone control model underlying our design and control of an AGV system. This model elaborates the structure of the road network (RN) and defines the vehicles' behavior when traveling on the road network. Then based on this model, a number of RN layouts are designed according to the specifications of a real ACT. With these settings, a traffic control strategy is proposed for the warranty of no inter-vehicle collisions and deadlocks. The traffic control provides some leeway for the vehicles to choose/change their routes online. Moreover, the control has small time complexity, hence is friendly for practical applications. Concerned about the system performance, a heuristic vehicle routing algorithm is developed which is oriented at minimizing the travel distance of the vehicles as well as the timespan for the container transportation. Incorporating with a task dispatching scheme of random nature, the performance of the AGV system with the set of proposed designs and controls is investigated by computer simulations. Finally, some discussions and open problems for future work are given.

The rest of the paper will be organized as the following. In Section 2, we formulate the zone control model. Section 3 is devoted to the layout design of the road network for an ACT. The traffic control strategy and routing algorithm are presented in Section 4. The performance evaluation of the AGV system will be done in Section 5. Finally, we make some concluding remarks and point out some directions for further research in Section 6.

* The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. INFSO-ICT-223844.

2. ZONE CONTROL MODEL

The workspace of the AGVs is divided into non-overlapping zones, and the entrance of a zone is strictly controlled. This strategy eases the avoidance of inter-vehicle collisions by demanding that each zone can be occupied by at most one vehicle. Another reason that makes it the favorite of most applications is that it is simple to install and easy to expand. Our zone control model described below contains two components: the structure of the road network and the behavior mode of a vehicle when it follows a route consisting of a sequence of zones.

2.1 Building blocks of the road network

The road network is composed of lanes, crossings and depots.

Lane and zone A lane is physically a road segment on which a vehicle can move with predefined direction. The direction that vehicles are allowed to move along on a lane is called the direction of the lane. A lane is composed of a finite sequence of zones which have to be passed in order when a vehicle is moving along it. Particularly, we call the first zone and the last zone of a lane the *starting zone* (SZ) and *ending zone* (EZ) of the lane respectively.

Every *depot* is modeled as a zone that can accommodate any number of vehicles. Each depot is affiliated with at least one *in-lane* and one *out-lane*. An in-lane (resp. out-lane) of a depot is a lane with the direction that allows a vehicle moving inwards (resp. outwards from) the depot.

We use $\mathcal{C}$ to denote the set of all zones in the system respectively.

Crossing A crossing is physically a junction area connecting multiple lanes. Specifically, each crossing is attached with a set of *in-lanes* and a set of *out-lanes*. An in-lane (resp. out-lane) of a crossing allows a vehicle to move towards (resp. away from) the crossing. The EZ of any in-lane of a crossing is named an *at-crossing zone* of the crossing. The zones that are not at-crossing zones are called *off-crossing zones*. Note that any depot is an off-crossing zone as it is not on any lane. Let $\mathcal{R}$ and $\mathcal{A}$ be the sets of all crossings and the set of all at-crossing zones respectively.

Neighboring zone The concept of neighboring zone characterizes immediate zone-to-zone connections. The set of neighboring zones of each depot includes the SZs of all its out-lanes. The neighboring zone of a non-EZ zone c on a lane is the adjacent zone of c on the lane with respect to the direction of the lane. The neighboring zone of the EZ of any in-lane of a depot is the depot. The neighboring zones of the EZ of an in-lane of a crossing (i.e. an at-crossing zone) are the SZs of all the out-lanes of the crossing. We use Υ_c to denote the set of neighboring zones of zone c . (One can see from Assumption 1 that neighboring zones are defined for any zone in $\mathcal{C}$.)

Conflicting zone pairs at a crossing For each crossing $r \in \mathcal{R}$, let $\mathcal{P}_r$ be a set of ordered pairs of zones, $\mathcal{P}_r = \{(c_1, c_2) : c_1 \text{ is an at-crossing zone of } r, c_2 \in \Upsilon_{c_1}\}$. Indeed, $\mathcal{P}_r$ characterizes all possible ways to pass the crossing r .

For each crossing $r \in \mathcal{R}$, with $\mathcal{P}_r \neq \emptyset$, and each zone pair $(c_1, c_2) \in \mathcal{P}_r$, there is a subset of $\mathcal{P}_r$, denoted by $\mathcal{X}_{(c_1, c_2)}^r$, which is called the set of conflicting zone pairs of (c_1, c_2) at the crossing r . As an example, in Figure 4, $\mathcal{P}_r = \{(c_1, c_3), (c_1, c_4), (c_1, c_5), (c_2, c_3), (c_2, c_4), (c_2, c_5)\}$ and $\mathcal{X}_{(c_1, c_5)}^r = \{(c_2, c_3), (c_2, c_4), (c_2, c_5)\}$, where r denotes the left crossing. To avoid inter-vehicle collision, it should be forbidden that two vehicles pass a crossing simultaneously via conflicting zone pairs.

We need the following assumption on the layout of the road network.

Assumption 1. (a) Each lane is an in-lane (out-lane) of either a unique crossing or a unique depot; (b) each lane has at least two zones; (c) any in-lane of a crossing has at least one neighboring lane.

2.2 Vehicle routes, states and events

We consider an AGV system with N vehicles, which is denoted by the set $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$. A finite sequence of zones $a_1^i, a_2^i, \dots, a_{e_i}^i$, $e_i \in \mathbb{N}$, is called a *route* of $v_i \in \mathcal{V}$ if $a_{p+1}^i \in \Upsilon_{a_p^i}$ for any $p \in \mathbb{Z}_1^{e_i-1}$ (For any $m, n \in \mathbb{N}$, $\mathbb{Z}_m^n$ is defined to be $\{m, m+1, \dots, n\}$ if $n \geq m$ and $\emptyset$ if $n < m$). There are only two types of vehicle states: being *in* a_p^i or *moving from* a_p^i to a_{p+1}^i . It is assumed that, at the initial time t_0 , v_i is in a_1^i .

Given the route, the event-driven behavior of the vehicle v_i can be sketched as follows. At a certain time when v_i is in a_p^i , $p \in \mathbb{Z}_1^{e_i-1}$, it triggers a “intend to leave” event. For the sake of collision and deadlock avoidance, the vehicle is required to check some traffic rules (to be specified in Section 4) to make a “go ahead” or “stop” decision. If the vehicle will be permitted to go ahead, it triggers a “leave” event and changes its state to be moving from a_p^i to a_{p+1}^i ; otherwise it has to stop but will continually trigger the “intend to leave” event for leaving the zone a_p^i . In the former case, at some time later on, after the vehicle has completely entered a_{p+1}^i , it will trigger an “arrive at” event, and changes its state to be in a_{p+1}^i . The process above keeps iterating and terminates if v_i is in $a_{e_i}^i$.

We say that a vehicle *arrives at*, *intends to leave*, or *leaves* some zone at time $t \geq t_0$ if the vehicle triggers the corresponding event at t . Note that a vehicle switches its state only when it arrives at or leaves a zone, and must be in the zone when it intends to leave a zone. A vehicle is called an at-crossing vehicle when it is in some at-crossing zone.

3. LAYOUT DESIGN OF THE ROAD NETWORK

3.1 Quayside Container Transport at an ACT

In this work, we consider a container handling scenario in which quay cranes (QCs) and yard stackers (Ys) are in charge of the container collection operations in the quay area (QA) and yard area (YA); and a team of AGVs are used for shuttling the containers between the two areas. The operation is illustrated in Figure 1. Specifically, in discharging a vessel, a couple of QCs take the containers off the vessel and put them in the associated container

Fig. 1. Quayside container transport at an automated container terminal

buffers. These containers will be later on picked up by some AGVs (e.g. straddle carriers or prime movers), and transported across the transportation area (TA) to the container buffers of specified YSs in the yard area. There the containers will be put in container stacks by the YSs. The other way around, in loading a vessel, the containers are collected from certain yard stacks by the YSs and transported to designated QCs by the AGVs. In this case study, the workspace of the AGVs is considered as the combined area of the QA, TA and YA.

Now suppose that a road network defined as in Section 2 is built over the workspace, with the buffer of each QC or YS modeled as one or more zone(s) (See also Subsection 3.2). For simplicity, these zones are said to be the zones of the QC or YS. Then based on the working scenario described above, each AGV in operation can be seen as being assigned a sequence of tasks, where a task of a vehicle means an ordered pair of zones, say (c_1, c_2) . We call the former zone c_1 and the latter c_2 the *source zone* and the *destination zone* of the task. In practice, tasks are assigned dynamically rather than fixed offline: after a vehicle arrives at the destination zone (normally a zone of a QC or YS) of the k th task and finishes a container loading/discharging process, it will be given the destination zone for the $k+1$ th task (the destination zone of the k th task becomes the source zone of the $k+1$ th task).

3.2 Layout Design of the Road Network

In this case study, we consider a typical large container terminal with a TA of the dimension $2000\text{m} \times 40\text{m}$; and 20 QCs and 66 YSs distributed on the sea side and yard side respectively. In Figure 2, we illustrate the workspace of the AGV system.

Fig. 2. Schematic representation of the workspace

Here, without any loss of generality, we assume that there is only one depot, with one in-lane and one out-lane, which is located outside the workspace (but connected to the TA by its in-lane and out-lane), and hence not a subject of our layout design. For this reason, in the remainder of this subsection, by zones we mean non-depot zones.

Geometrical properties of the zones First we fix the shape of the zones to be rectangular because the rectangular shape of the vehicles. We call the dimension of the edge of a zone along which a vehicle moves the length of the zone; the dimension of the other edge the width. The approximate geometric and kinematic specifications of the AGVs are listed in the table below. (These are the typical values used in this paper, and can be easily adjusted depending on actual situations.)

Geometric and kinematic characteristics of AGVs				
Length	Width	Max speed	Max acc.	Max dec.
10m	4m	7m/s	$0.5\text{m}^2/\text{s}$	$2.5\text{m}^2/\text{s}$

The basic principle for determining the length of a zone is that, for safety reasons, the body of a vehicle must be completely inside a zone when the vehicle is in the zone. As shown in Figure 3, in order to minimize the length of a zone, we locate the sensor for triggering the “arrive at” event close to one end of the zone (Line A), and put the sensor for triggering the first “intend to leave” event nearby (Line B). Recall that a vehicle may not be allowed by the traffic rules to move on to the next zone on its route, this requires that the zone must be long enough so that if a vehicle starts braking at Line B, its body is still completely inside the zone when it fully stops. In the case that a vehicle is in its maximum speed (see the table below) when triggering the first “intend to leave” event, it needs approximately 11 meters for a full brake. Hence the length of a zone is finalized to be 22m, a little more than the sum of the length of a vehicle and the braking distance. Since we ignore the side sliding of the vehicles, the width of zones can be safely put approximately at 7m.

Fig. 3. Event triggering of a vehicle: the small circle on the vehicle indicates the reference position for event triggering

Layout design for the quay area and yard area For simplicity, the road network has the same layout at each QC or YS, which is illustrated in Figure 4. The buffer of a QC or YS is modeled by the SZs on three different lanes in between two crossings. The 2 in-lanes of one of the two crossings (left one in Figure 4) allow vehicles to drive off the TA and towards the QC or YS; while the 2 out-lanes of the other crossing lead the vehicles back to the TA.

Layout design for the transportation area In view of the geometric characteristics of the TA, we focus on the layouts with straight adjacent roads that are parallel to the x -axis in Figure 2. A road is indeed composed of multiple lanes, joined by crossings, on the same level with respect to the y -axis. See Figure 5 for an example of such layouts.

Fig. 4. Road network at a QC or YS

For simplicity, each crossing is chosen to be rectangular. In order to avoid collision between a vehicle passing some crossing and another vehicle waiting at the same crossing. Each crossing should be able to accommodate the whole body of a vehicle. Considering the turning radius of the vehicles, the minimum x -dimension of the crossings is 15m. Note that adopting crossings with larger x -dimension increases the time for vehicles to pass them, and hence in general increases the waiting time of at-crossing vehicles. Therefore we fix the length of each crossing to be its minimum. The y -dimension of a crossing depends on the layout design and its location in the TA.

As in practice the y -dimension of the TA cannot exceed 40m, it allows at most 6 parallel roads. Based on this restriction, four types of layouts for the TA will be discussed:

- (T1) 4 parallel roads and small crossings.
- (T2) 4 parallel roads and big crossings.
- (T3) 6 parallel roads and small crossings.
- (T4) 6 parallel roads and big crossings.

An example of each type of layout is shown in Figures 5-8. (In fact, we have 19 designs of all the four types, but cannot fit them all into the limited space here.)

A small crossing is defined as a crossing which connects two adjacent parallel roads (See Figure 5). A big crossing is a crossing which connects more than two parallel roads. One advantage of small crossings is that only a small number of AGVs need to wait at the crossing while some other vehicle is passing the crossing. The main disadvantage of using small crossings is that an AGV has to pass a relatively large number of crossings for a QC-YS or YS-QC task. On the other hand, the use of big crossings might increase the waiting time of the AGVs at the crossings due to possibly more vehicles competing for the crossings; but in general, compared with using small crossings, it reduces significantly the distances of the transportation tasks by providing direct channels linking the QA and the YA. The latter point is illustrated in Figures 5 and 6 by two routes (in dashed lines) that serve the same purpose for a vehicle to cross the TA.

In each design, the crossings with vertical arrows in the figure set up connections between the part of the road network in the TA and those in the QA and the YA. In detail, each crossing with an upward (resp. downward) arrow, called a TA-YS (resp. TA-QC) crossing is linked with the in-lanes and out-lanes of two neighboring YSs (resp. one QC), i.e. it allows a vehicle to move to two YSs (one QC) from the TA and vice versa by passing it. In view of Figure 4, this implies that every TA-YS (resp. TA-QC) crossing has 4 out-lanes and 4 in-lanes in connection with

the YA (resp. 2 out-lanes and 2 in-lanes in connection with the QA). Note that the number of the TA-YS crossings, which is $66/2 = 33$ (66 is the number of YSs), already reaches its maximum value due to the restriction that any lane must have at least two zones (see Assumption 1(b)).

Fig. 5. An example of layout type T1

Fig. 6. An example of layout type T2

Fig. 7. An example of layout type T3

In the layouts consisting of 4 parallel roads, all lanes in the 2 upper roads have the same direction which is opposite to the common direction of the lanes in the 2 lower roads. Analogously, in the case of 6-road layouts, the direction of all lanes in the 3 upper roads is the opposite to that of the lanes in the 3 lower roads. The thinking behind these choices of lane directions is to make the AGVs brake as little as possible when traveling across the TA. This can be illustrated by the examples in Figure 9.

Fig. 8. An example of layout type T4

Fig. 9. Different choices of lane directions

4. TRAFFIC CONTROL AND ROUTING ALGORITHM

4.1 Traffic Control

The aim of the traffic control is to prevent inter-vehicle collisions and deadlocks. This is an important issue that has attracted intensive research effort [Evers and Koppers, 1996, Rajeeva et al., 2003, Steenken et al., 2004, Vis, 2006, Kim et al., 2006]. See also Fanti et al. [1997], Yeh and Yeh [1998], Reveliotis [2000], Wu and Zhou [2005] for related studies. Compared with most approaches in literature, our traffic control strategy has two merits: (1) it allows the vehicles to dynamically make/change their routes online; and (2) it is time-efficient, the complexity linearly depends on the number of vehicles.

Before stating our traffic control rules, we introduce some instrumental notions and assumptions on collision avoidance and initial condition.

Zone state Each zone can have two states: being “occupied” and “available”. The depot zone is always available. A non-depot zone c is said to be occupied by a vehicle, if the vehicle is either (a) moving from some zone to c ; (b) in c ; or (c) moving from c to some zone. A non-depot zone c is available if it is not occupied by any vehicle. It is clear that the states of zones change only when the states of vehicles change.

To ease presentation, we say that a zone is *white* if it is available; and that a non-depot zone is *black* if it is occupied by only one vehicle which is either in it or moving from some other zone to it.

Next zone and black cycle Let $c \in \mathcal{C} \setminus \mathcal{A}$, then the set of the next zones of c is identical with Υ_c (the neighboring zone set of c , see Section 2). Let $c \in \mathcal{A}$. If there is some vehicle in c or moving from c to c^* at t , then c^* is a next zone of c at t ; otherwise, c has no next zone at t .

A black cycle is a sequence of zones $c_1, c_2, \dots, c_n, c_1$, $n \geq 2$, that are all distinct and black, where c_{i+1} is a next zone of c_i for any $i \in \mathbb{Z}_1^{n-1}$ and c_1 is a next zone of c_n .

Assumption 2. For any v_i, v_j , $i, j \in \mathbb{Z}_1^N$, $i \neq j$, v_i does not collide with v_j at time t if either of the following conditions is true:

- (a) v_i is in c_1 and v_j is in c_2 ;
- (b) v_i (v_j) is in c_1 and v_j (v_i) is moving from c_2 to c_3 ;
- (c) v_i is moving from c_1 to c_2 and v_j is moving from c_3 to c_4 ;

where the non-depot zones in each of the three cases are distinct; and in (c) $(c_3, c_4) \notin \mathcal{X}_{(c_1, c_2)}^r$ for any $r \in \mathcal{R}$.

Assumption 3. At time t_0 , (a) each vehicle is in some zone (called the *initial zone* of the vehicle), (b) there cannot be two vehicles occupying a same non-depot zone, and (c) there is no black cycle.

Traffic rules In the statements of traffic rules below, we assume that vehicle v_i is with the route $a_1^i, a_2^i, \dots, a_{e_i}^i$, $e_i \in \mathbb{N}$, where a_1^i is the initial zone of the vehicle. To concisely represent the running status of a vehicle, we define the following function $h_v(i, t) : \mathbb{Z}_1^N \times [t_0, \infty) \rightarrow \mathcal{C} \times \mathcal{C}$ as follows:

$$h_v(i, t) = \begin{cases} a_p^i a_p^i, & \text{if } v_i \text{ is in } a_p^i \text{ at } t \wedge w_i(t) = p; \\ a_p^i a_{p+1}^i, & \text{if } v_i \text{ is moving from } a_p^i \text{ to } a_{p+1}^i \text{ at } t \wedge w_i(t) = p, \end{cases}$$

where $w_i(t) = p$ means that v_i has been successively in $a_1^i, a_2^i, \dots, a_p^i$ over $[t_0, t]$.

Firstly, for the purpose of the collision avoidance it is necessary to impose a rule that does not allow a vehicle to head forward when it is blocked by another one. Indeed this minimum requirement becomes our rule applied to off-crossing zones.

Rule 1: Traffic Rules for Off-Crossing Zones

Suppose that v_i intends to leave an off-crossing zone a_p^i at t , with $h_v(i, t) = a_p^i a_p^i$. Then over some time window $\mathcal{T} = [t, t + \epsilon]$, $\epsilon > 0$, if the vehicle finds the zone a_{p+1}^i available over $[t, t + \epsilon]$, then the vehicle changes its state at $t + \epsilon$ to be moving from a_p^i to a_{p+1}^i ; otherwise the vehicle will again intend to leave a_p^i at some time $t' > t + \Delta$, where $\Delta > \epsilon$ is some positive constant.

Now, note that applying Rule 1 to every zone cannot prevent deadlocks of the AGV system. This motivates us to put forward a bit more sophisticated traffic rule for at-crossing zones.

Before stating our traffic rule for at-crossing zones, we introduce a global *crossing token* to guarantee that there is at most one at-crossing vehicle which can change its state at any time. In fact, it is required here that each at-

crossing vehicle must request for the crossing token before triggering an “intend to leave” event and hold the token exclusively over the time slot (the interval $\mathcal{T}$ in Rule 2 below) it opens to check with the at-crossing traffic rule.

Rule 2: Traffic Rules for At-Crossing Zones

Suppose that v_i intends to leave an at-crossing zone a_p^i of the crossing r at t , with $h_v(i, t) = a_p^i$. Then over a time window $\mathcal{T} = [t, t + \epsilon]$, $\epsilon > 0$, v_i must hold the global crossing token; and if the vehicle finds that

- (a) the zone a_{p+1}^i is available over $[t, t + \epsilon]$;
- (b) there is no such vehicle v_j , $j \neq i$, with $h_v(j, t + \epsilon) = a_q^j a_{q+1}^j$, where $(a_q^j, a_{q+1}^j) \in \mathcal{X}_{(a_p^i, a_{p+1}^i)}^r$;
- (c) to change its state to be moving from a_p^i to a_{p+1}^i at $t + \epsilon$ does not lead to a black cycle containing a_{p+1}^i ;

then the vehicle changes its state at $t + \epsilon$ to be moving from a_p^i to a_{p+1}^i ; otherwise the vehicle will again intend to leave a_p^i at some time $t' > t + \Delta$, where $\Delta > \epsilon$ is some positive constant. In both cases, the vehicle releases the crossing token after making the decision.

Note that to satisfy the token-holding requirement, a token assignment algorithm is necessary as there can be multiple vehicles waiting for the token simultaneously to trigger their “intend to leave” event.

Here we introduce a token assignment algorithm, where a central controller maintains an index list of the “starving” vehicles that have sent their token requests but not been granted it yet. The position of the vehicle indices in the list are ordered by their degrees of starvation: the one waiting for the longest time is on the top, and the one with the shortest wait is at the bottom. Among the vehicles with the same waiting time, the order is immaterial. The rules for the token assignment are as follows.

Crossing Token Assignment Algorithm

- Step 1: Begins with assigning the token to the 1st vehicle from the top.
 - Step 2: If the token is released by the n th (not the last) vehicle without a state change of the vehicle, then assign the token to the $n + 1$ th vehicle. If the token is released with a state change of the vehicle, then delete the n th (can be the last) vehicle from the list and assign the token to the 1st vehicle.
 - Step 3: If the last vehicle in the list releases the token without a state change, assign the token to the 1st vehicle in the list.
-

One can easily show that this algorithm feeds the token to a vehicle in finite time after it requests it. The motivation behind Step 2 is to balance the “degrees of starvation” of the competing vehicles: the state change of an at-crossing vehicle may offer a chance for some others to leave their current zones; and once such a possibility appears, the token is fed to the vehicle that needs it most urgently. This heuristic helps reduce the overall lateness (namely the *tardiness* specified in Section 5) of the container deliveries.

The traffic rules (i.e. Rule 1 and Rule 2) together with the token assignment algorithm constitute our traffic control for the AGV system. The main conclusion we can draw from applying this traffic control is as follows. Suppose Assumptions 1, 2 and 3 hold. Then it is ensured that (1) each vehicle complete any route in finite time if the last zone of the route is the depot; and (2) no inter-vehicle collision occurs. [Li et al., 2010]

Now suppose that, in the operation of the AGV system of an ACT, vehicle v_i is assigned a total of $M_i \in \mathbb{N}$ tasks. We say that a zone sequence $b_{k,1}^i, b_{k,2}^i, \dots, b_{k,n_k}^i$ is a route for the k th task, $k \in \mathbb{Z}_1^{M_i}$, of vehicle v_i if $b_{k,1}^i$ and b_{k,n_k}^i are the source zone and the destination zones of the task respectively, and $b_{k,p+1}^i \in \Upsilon_{b_{k,p}^i}^i$ for any $p \in \mathbb{Z}_1^{n_k-1}$. As remarked in Subsection 3.1, we also have $b_{k+1,1}^i = b_{k,n_k}^i$ for any $k \in \mathbb{Z}_1^{M_i-1}$. Assuming that $b_{M_i,n_{M_i}}^i$ is a depot, if we glue the routes for all the tasks together and see the integrated zone sequence $b_{1,1}^i, \dots, b_{1,n_1}^i(b_{2,1}^i), \dots, b_{2,n_2}^i(b_{3,1}^i), \dots, b_{M_i,n_{M_i}}^i$ as a route of v_i , then by applying the traffic control above, we can make sure that each vehicle will eventually finish all the given tasks without colliding with others.

An appealing advantage of the proposed traffic control is that it offers a large extent of freedom in routing the AGVs. In fact, the route for each task of a vehicle can be determined online rather than necessarily fixed beforehand. Specifically, before vehicle v_i arrives at the p th ($p \geq 2$) zone of its route for the k th task $b_{k,p}^i$, it may select any zone in $\Upsilon_{b_{k,p}^i}^i$ as $b_{k,p+1}^i$ provided that the dynamically established route will end up with the destination zone of the task.

4.2 Routing algorithm

As mentioned in the previous subsection, before a vehicle arrives at some zone c it has some freedom to choose the next zone on its route out of the neighboring zones of c . But it is easy to see that if c is an off-crossing zone, then c has only one neighboring zone. Therefore, the only case of interest is when c is an at-crossing zone. Roughly speaking, the routing algorithm introduced here ensures that the vehicle always chooses a zone such that it will be closer to the destination of its current task at the resulting next crossing. As a direct consequence, this guarantees the vehicle reach the destination in finite zones. Beyond this basic property, the algorithm also intends to “smooth” the motion of the vehicles as much as possible.

Now suppose that c_d is the destination zone of the current task of some vehicle, which may be some zone of a QC, a YSs or the depot. Then before arriving at some at-crossing zone c of crossing $r \in \mathcal{R}$, the vehicle needs to select a zone out of the zones in $\Upsilon_c = \mathcal{S}_r := \{c_s^i : l_i \in \mathcal{O}_r\}$, where $\mathcal{O}_r = \{l_i\}$ is the set of out-lanes of crossing r and c_s^i denotes the SZ of l_i . For any zone c^* , we use $d_m(c^*, c_d)$ and $x_m(c^*, c_d)$ to represent the length of the shortest route(s) between zone c^* and c_d and the smallest number of crossings a vehicle has to pass on a way running from c^* to c_d .

Before we state our routing algorithm, define two zone sets:

$$\Gamma_1 = \{c_s^i \in \mathcal{S}_r : d_m(c_s^i, c_d) < d_m(c, c_d)\},$$

$$\Gamma_2 = \{c^* \in \Gamma_1 : c^* \text{ is available}\};$$

where c_e^i is the EZ of the lane l_i (By assumption 1(a), c_e^i is either an at-crossing zone or the EZ of the in-lane of the depot), and let

$$\Gamma_3 = \begin{cases} \{c^* \in \Gamma_2 : c^* = \operatorname{argmin} d_m(c^*, c_d)\}, & \text{if } \Gamma_2 \neq \emptyset; \\ \{c^* \in \Gamma_1 : c^* = \operatorname{argmin} d_m(c^*, c_d)\}, & \text{otherwise.} \end{cases}$$

Online Vehicle Routing Algorithm

- Step 1: If c_d is on some lane $l_k \in \mathcal{O}_r$, then choose the zone c_s^k ; otherwise go to Step 2.
- Step 2: If $\#\Gamma_3 = 1$, then choose the single zone in Γ_3 ; otherwise go to Step 3.
- Step 3: Let $\Gamma_4 = \{c^* \in \Gamma_3 : c^* = \operatorname{argmin} x_m(c^*, c_d)\}$. If $\#\Gamma_4 = 1$, then choose the single zone in Γ_4 ; otherwise go to Step 4.
- Step 4: If there is a zone c^* in Γ_4 to reach which the vehicle does not need a turn, then c^* is selected; otherwise choose randomly a zone from Γ_4 .

Remark 4. In each layout of the road network designed in Subsection 3.2, the set Γ_1 cannot be empty; and such a c^* in Step 4, if exists, must be unique.

Note that to implement the routing algorithm above, one needs to maintain for each crossing r and each possible destination zone c_d the data sets $\{d_m(c, c_d) : c \text{ is any at-crossing zone of } r\}$, $\{d_m(c_s^i, c_d), x_m(c_s^i, c_d), d_m(c_e^i, c_d) : \text{for any } l_i \in \mathcal{O}_r\}$. But for each given layout, these data can be generated and stored offline before the system starts running. Therefore, the time complexity of the proposed routing algorithm is very small.

5. SIMULATION STUDIES

In this section, via computer simulations, we investigate the performance of the AGV systems with the designs and controls presented in the preceding content.

5.1 Task dispatching scheme

Here we introduce a simple stochastic task dispatching scheme that is used to drive the simulations. We emphasize that an in-depth discussion on this topic is beyond the scope of the paper. See Steenken et al. [2004] and Stahlbock and Voß [2008b] for recent progress in the research on this subject.

There are three types of tasks for each AGV:

- (1) traveling from the depot to a zone of a QC or YS (first task only);
- (2) traveling across the TA, i.e., from a zone of a QC to a zone of a YS or the other way around;
- (3) traveling from a zone of a QC or YS to the depot (last task only).

For the tasks of the second type, the source and destination stations are not arbitrarily chosen. Instead, according to a typical container transshipment scenario, the 20 QCs and 66 YSs are divided into 5 groups respectively. The groups of the QCs are disjoint, each of which consists

of 4 QCs; while the groups of the YSs are overlapping with each group containing 24 YSs. For each vehicle, the source and destination zones for any task of the second type are randomly selected respectively from the zones of the related QC and YS groups (see Figure 10).

In addition, for the tasks of types (1) and (2), after a vehicle arrives at the destination zone of each task, it will stay stationary in the zone for 30 seconds for a container loading or discharging process before being assigned a new task.

Fig. 10. Task dispatching scheme

5.2 Performance measures

Several performance measures are defined here in different aspects of concern. The first measure is related to the throughput of the terminal, called “average tasks performed per hour” (ATP), which is defined as

$$ATP = \frac{3600}{ATT}, \quad (1)$$

where ATT denotes the “average task time” calculated by

$$ATT = \frac{\sum_{i=1}^N \sum_{j=1}^{M_i} T_j^i}{\sum_{i=1}^N M_i}, \quad (2)$$

with T_j^i the time (in second) for vehicle v_i to complete its j th task. (Recall that N is the number of vehicles; and M_i is the number of tasks for vehicle v_i .)

From an energy cost point of view, another performance measure of interest is the “average travel distance” (ATD) whose definition mimics that of ATT with T_j^i replaced by D_j^i , the travel distance for the j th task of v_i .

The last two performance measures, namely “average tardiness” (ATR) and “standard deviation of the tardiness” (STDTR), indicate how much are the motions of the AGVs delayed by the traffic. The formula for computing the ATR has the same form as (2) but with T_j^i substituted by

$$D_j^i = T_j^i - \text{min time required to complete task } j \text{ of } v_i$$

Here the second term on the right hand side is the time for v_i to complete the task via the shortest route while assuming v_i the only vehicle running in the workspace.

The STDTR is computed by

$$STDTR = \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^{M_i} (D_j^i - ATR)^2}{\sum_{i=1}^N M_i}}. \quad (3)$$

5.3 Simulation results

Because of the page limit, we will only show the results for the layouts illustrated in Figures 5-8. (In fact, we have done simulations for all the 19 designs, and it turned out that the four selected here are the best of their respective types.)

We run simulations with $N = 20, 30, \dots, 120$ and $M_i = 50$ for any $i = \mathbb{Z}_1^N$. In order to obtain adequate precision of analysis, we repeat the simulations for each layout for 20 times, and take the average value for each performance measure. The results are depicted in Figures 11-14, where the layouts in Figures 5,6,7 and 8 are named Layouts 1,2,3 and 4 respectively.

Fig. 11. Average performed tasks per hour

Fig. 12. Average tardiness for a task

Fig. 13. Standard deviation of the tardiness

Fig. 14. Average traveled distance for a task

It can be seen that Layout 2, i.e. the layout with four roads and big crossings, remains the best choice for this wide range of the size of the vehicle team. This result can be roughly explained as follows: As mentioned in Subsection 3.2, the use of small crossings may have less vehicles waiting at the crossings in the TA (not much less since simultaneous passings of a big crossing by two vehicles are not necessarily conflict with each other). However, in average, the travel distance of the vehicles in this case is considerably larger than that in the case of using big crossings (see Figure 14). The use of six roads, compared with four roads, renders more space (lanes, zones) for vehicles' movement. But this benefit is offset by increased travel distances and number of vehicles competing for crossings.

In addition, using Layout 2, one can see from Figures 12 and 13 moderate ATR and STDTR for a large team of operational vehicles. This observation shows that by the proposed design and control strategies, if the vehicles in the system are dispatched in a random and uniform manner, then even for a busy operation (1) the flow of the vehicles is quite smooth, i.e. the motion of the vehicles are not much interfered by the traffic; and (2) one can estimate for each task the arriving time of the vehicle performing the task with an acceptable error.

6. CONCLUSIONS AND DISCUSSIONS

In this paper, we give a solution to the design and control of the AGV system for an automated container terminal. The popular zone control approach is used here to ease the traffic management. Various road network layouts are designed based on the practical dimension of a container terminal. A traffic control scheme as well as a routing algorithm are developed. Computer simulations demonstrate the efficiency of our results in terms of some defined performance measures.

For an interesting comparison, we also did computer simulations in which the AGVs are free ranging (no road network is used), and the inter-vehicle collision avoidance are realized by the so called potential field based approaches [Whitcomb et al., 1992]. Our findings can be summarized as follows. With a small amount of AGVs ($N < 20$), the two methods are comparable: AGVs have similar average task time and travel distances (the free ranging case can be a bit better). However, for a relatively large vehicle

fleet ($N > 30$), our method outperforms its free ranging counterpart; and the differences of the performance measures stand out as the size of the fleet increases. Based on our observations, the main reason for this result is that a large number of the nearly head-to-head motion conflict resolutions, due to the narrowness of the transportation area, heavily slow down the vehicles in the free ranging case.

There are several possible directions for further research. First, in our traffic control strategy, it is required that each lane of the road network must contain at least two zones. To somehow weaken this restriction may be important for some applications where the workspace of the AGVs is very limited. In addition, one could think of a relaxation of the token-holding requirement in the traffic control scheme so that multiple vehicles can leave different at-crossing zones simultaneously, and hence the performance of the AGV system can be improved. As far as we could tell, using local crossing tokens, instead of a global one required in this work, and adding inter-crossing communications may lead to a successful trial.

The designs of the layout of the road network and the routing algorithm are vital to the performance of the AGV system. The designs presented in this paper are partially heuristic with the help of intuitions and practical experiences. Better results could be obtained by setting up and solving formal optimization problems. Along this line of thinking, the balance between the optimality and real-time applicability of the solutions become critical for practical applications.

In this work, the tasks of the AGVs are assigned in a purely random manner. We predict a room for performance improvement in the future if a sophisticated task dispatching strategy could be incorporated, although we have seen only a moderate tardiness for a typical number of operational AGVs with our current strategy. Such a dispatching strategy is desired to work collaboratively with the routing algorithm, and utilize the feedback of the real-time traffic flow information.

Another interesting topic is to equip the AGV system with some fault-tolerant mechanisms so that it can still run safely and smoothly with unexpected disturbances, such as abrupt changes of container delivery schedule and the appearance of obstacles in the workspace, or even with some failures, such as the breakdowns of vehicles or cranes.

REFERENCES

- E. K. Bish, F. Y. Chen, Y. T. Leong, B. L. Nelson, J. W. Cheong Ng, and D. Simchi-Levi. Dispatching vehicles in a mega container terminal. *OR Spectrum*, 27:491–506, 2005.
- D. Briskorn, A. Drexler, and S. Hartmann. Inventory-based dispatching of automated guided vehicles on container terminals. *OR Spectrum*, 28:611–630, 2006.
- M. B. Duinkerken, J. A. Ottjes, and G. Lodewijks. Comparison of routing strategies for agv systems using simulation. In *Proc. of the 2006 Winter Simulation Conference*, Monterey, Canada, 2006.
- J. J. M. Evers and S. A. J. Koppers. Automated guided vehicle traffic control at a container terminal. *Transportation Research Part A: Policy and Practice*, 30: 21–34, 1996.
- M. P. Fanti, B. Maione, S. Mascolo, and B. Turchiano. Event-based feedback control for deadlock avoidance in flexible production systems. *IEEE Trans. on Robotics and Automation*, 13:347–363, 1997.
- K. H. Kim and J. W. Bae. A dispatching method for automated guided vehicles to minimize delays of container-ship operations. *International Journal of Management Science*, 5:1–25, 1999.
- K. H. Kim, S. M. Jeon, and K. R. Ryu. Deadlock prevention for automated guided vehicles in automated container terminals. *OR Spectrum*, 28:659–679, 2006.
- M. Lehmann, M. Grunow, and H.-O. Günther. Deadlock handling for real-time control of agvs at automated container terminals. *OR Spectrum*, 28:631–657, 2006.
- Q. Li, J. T. Udding, and A. Y. Pogromsky. Modeling and control of the AGV system in an automated container terminal. In *Proc. of AsiaMIC 2010*, Phuket, Thailand, to appear, 2010. (available at <http://w3.wtb.tue.nl/en/people/pages/?script=showabstract.php&outid=12192>).
- V. D. Nguyen and K. H. Kim. A dispatching method for automated lifting vehicles in automated port container terminals. *Computers & Industrial Engineering*, 56: 1002–1020, 2009.
- L. M. Rajeeva, H. G. Wee, W. C. Ng, C. P. Teo, and N. S. Yang. Cyclic deadlock prediction and avoidance for zone-controlled agv system. *International Journal of Production Economics*, 83:309–324, 2003.
- S. A. Reveliotis. Conflict resolution in agv systems. *IIE Transactions*, 32:647–659, 2000.
- R. Stahlbock and S. Voß. Vehicle routing problems and container terminal operations - an update of research. In S. Raghavan B. Golden and E. Wasil, editors, *The Vehicle Routing Problem - Latest Advances and New Challenges*, pages 551–589. Springer, Berlin, 2008a.
- R. Stahlbock and S. Voß. Operations research at container terminals: a literature update. *OR Spectrum*, 30:1–52, 2008b.
- D. Steenke, A. Henning, S. Freigang, and S. Voß. Routing of straddle carriers at a container terminal with the special aspect of internal moves. *OR Spectrum*, 15:167–172, 1993.
- D. Steenken, S. Voß, and R. Stahlbock. Container terminal operation and operations research - a classification and literature review. *OR Spectrum*, 26:3–49, 2004.
- I.F.A. Vis. Survey of research in the design and control of automated guided vehicle systems. *European Journal of Operational Research*, 170:677–709, 2006.
- L. L. Whitcomb, D. E. Koditschek, and J. B. D. Cabrera. Toward the automatic control of robot assembly tasks via potential functions: the case of 2-d sphere assemblies. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Nice, France, 1992.
- N. Q. Wu and M. C. Zhou. Modeling and deadlock avoidance of automated manufacturing systems with multiple automated guided vehicles. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35:1193–1202, 2005.
- M. S. Yeh and W. C. Yeh. Deadlock prediction and avoidance for zone-control agvs. *International Journal of Production Research*, 36:2879–2889, 1998.