PERFORMANCE OF TWO REAL TIME CONTROL STRATEGIES FOR AGV SYSTEMS: A CASE STUDY

M. Dotoli*, M. P. Fanti*

*Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari
Via Re David 200, Bari, 70125 Italy
Email: {dotoli, fanti}@poliba.it
Fax: +39 080 5963410

Keywords: Automated guided vehicle systems, real time control, coloured timed Petri nets, deadlock avoidance, discrete event simulation.

Abstract

Real time control of material handling devices is essential to guarantee efficiency and flexibility of automated manufacturing systems. This paper presents a performance based comparison of two control policies previously presented by one of the authors to avoid deadlock and collisions in zone controlled Automated Guided Vehicle Systems (AGVSs). Coloured Timed Petri Nets are used to model the dynamics of AGVSs and implement the control strategies stemming from the knowledge of the system state. Several simulations of an AGVS with varying fleet size show the effectiveness of one of the considered control strategies compared to the alternative policy.

1 Introduction

Efficient and effective real time control has basic importance to manage Automated Guided Vehicle Systems (AGVSs). AGVSs are flexible material handling devices suitable to improve the performance of Automated Manufacturing Systems (AMSs) [9]: the controller assigns route and velocity to vehicles and avoids collisions and deadlocks during the AGVs missions.

This paper makes use of a standard technique for vehicle management of AGVSs, i.e., zone control. More precisely, the guide-paths are separated into disjoint zones and deadlocks can occur when a set of AGVs competes for a set of zones detained by vehicles of the same set. In particular, to cope with deadlock in zone-controlled AGVSs, Fanti [1, 2, 3] proposed some deadlock avoidance policies based on digraph analysis. Such algorithms avoid not only deadlock states but also some peculiar conditions, known as “restricted deadlock” [4]. When a restricted deadlock occurs, the system is not in deadlock condition but some vehicles permanently remain in circular wait, partly because some of them are blocked and partly since the controller prevents them from moving. Following the approach in [1], the AGVS controller structure is composed of two levels: the path scheduler and the real time controller. The path scheduler is a higher control level selecting the paths to assign to the vehicles. On the other hand, the real-time controller has two functions: i) it validates the path proposed to a vehicle (path validation) so that each mission is completed without deadlock and restricted deadlock; ii) it validates the next zone in the path to prevent deadlocks and collisions, by enabling or inhibiting the AGV zone acquisition (zone validation).

This paper focuses on the specification of the real time controller. The AGVS structure and dynamics are described by a resource oriented Coloured Timed Petri Net (CTPN): tokens are vehicles and the path that each AGV has to complete provides the token colour. Moreover, associating a time concept to the Coloured Petri Net allows the investigation of the system performance. Indeed, we propose a performance based comparison of two control policies managing traffic in AGVSs previously presented by one of the authors. In particular, the paper considers the most flexible procedure to avoid deadlock and restricted deadlocks in AGVS based on digraphs presented in [1]. An alternative approach presented in [3] bases decisions on the results of a simulation run of the CTPN model, forecasting the AGVS behaviour. In order to compare the two approaches, an AGVS is adopted as a case study. Simulation evidences in the MATLAB-Stateflow environment [8] show that the policy introduced in [3] exhibits better performance indices than the control strategy presented in [1].

The paper is organised as follows. Section 2 describes the AGVS layout and the CTPN model. Section 3 recalls previous results on deadlock avoidance in AGVSs. In addition, Section 4 recalls the considered control policies and Section 5 describes the controller synthesis. Finally, Section 6 presents several simulation results for the case study and Section 7 draws the conclusions.

2 The AGVS model

2.1 The AGVS description

The case study adopted in this paper is an AGVS with a layout involving unidirectional and bi-directional guide-paths. The AGVS is divided into several disjoint zones and each zone can represent a workstation, an intersection of paths or a straight lane (see figure 1). Moreover, the AGVS includes a docking station where idle vehicles are parked. The set of zones of the AGVS is denoted by $Z=\{z_i\ i=1,\ldots,N_Z\}$, where z_i for $i=2,\ldots, N_Z$ represents a zone and z_1 denotes the docking station. In addition, $V=\{v_j: j=1,\ldots, N_V\}$ is the set of vehicles available in the system. Since each zone can accommodate only one vehicle at a time, zones z_i for $i=2,\ldots, N_Z$ have unit capacity. On the contrary, the docking station can accommodate all the vehicles and is modelled by zone z_1 with capacity equal to N_V. The AGVS shown by figure 1 connects six workstations (denoted in the figure by z_2, z_3, z_4, z_5 and z_6) and the docking station z1. Four additional zones denote the intersections of the paths with parts of lanes (z_7, z_8, z_9 and z_{10}). The paths z_1-z_7, z_1-z_{10}, z_{10}-z_9, z_7-z_8 and z_8-z_9 are bi-directional and the others are unidirectional. We describe the route **r**(v) assigned to the vehicle v∈V by the zones sequence **r**(v)=(z_i…z_j…z_1) that ends to the docking station. In the sequel, **rr**(v) denotes the residual route that v∈V has to visit to complete its travel starting from a certain system configuration: obviously, it is a subsequence of **r**(v).

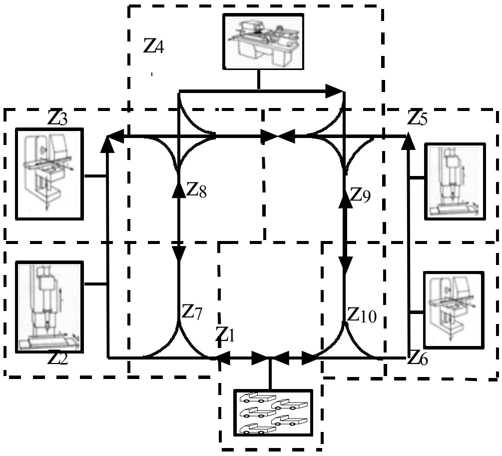

Figure 1. A zone-control AGVS.

2.2 The coloured timed Petri net model

This section recalls a modular method to build the coloured Petri net modelling the system layout and dynamics [2, 3]. We assume that the reader is familiar with Coloured Timed Petri Nets (CTPN) (see [6] for details).

A CTPN is an 8-tuple CTPN=(P, T, Co, Inh, $\mathbf{C}^+$, $\mathbf{C}^-$, Ω, M_0) where P is a set of places, T is a set of transitions, Co is a colour function defined from $P\cup T$ to a set of finite and not empty sets of colours [6]. Co maps each place p∈P to a set of possible token colours Co(p) and each transition t∈T to a set of occurrence colours Co(t).

In our model, a place $z_i \in P$ denotes the zone $z_i \in Z$ and a token in z_i represents a vehicle that is in zone z_i. The transition set T models the guide-paths between consecutive zones. Moreover, the set of arcs $(P\times T)\cup(T\times P)$ is defined as follows: if z_i and z_m are two consecutive zones in the AGVS, then transition t_{im} belongs to T and it is such that $t_{im}\in\bullet z_m$ and $t_{im}\in z_i\bullet$. To admit just one vehicle in each zone other than the docking station, there is an inhibitor arc between each place $z_i\in P$ with i≠1 and transition $t_{im}\in T$, i.e., $Inh(z_i,t_{im})=1$. More precisely, the inhibitor arc between a place $z_i\in P$ and a transition $t_{im}\in T$, implies that transition t_{im} can be enabled if z_i does not contain any token. Since z_1 can accommodate N_V tokens, there are no inhibitor arcs between z_1 and each $t_{i1}\in T$.

Having described the skeleton of the CTPN, it is necessary to model the AGVS behaviour and the travels of vehicles. Hence, each AGV v∈V is modelled by a coloured token and its token colour is <**rr**(v)>, where **rr**(v) is the residual path that the vehicle has to follow to reach the docking station. A marking M is a mapping defined over P so that $M(z_i)$ is a set of elements of $Co(z_i)$, also with repeated elements, i.e., a multi-set [6] corresponding to token colours in the place z_i. The state of the AGVS is represented by the marking of the CTPN, i.e., if $M(z_i)$=<**rr**(v)>, then vehicle v is in zone z_i and its colour <**rr**(v)> represents the sequence of zones that v has to visit starting from the current marking. Consequently, the colour domain of place $z_i\in P$ is: $Co(z_i)$={<**rr**> where **rr** is a possible sequence of zones and z_i is the first zone of **rr**}.

Moreover, Co associates with each transition t_{im} a set of possible occurrence colours: $Co(t_{im})$={<**rr**> such that **rr** is a possible sequence of zones and z_i and z_m are respectively the first and the second element of **rr**}. Here, the CTPN is represented by the $|P|\times|T|$ pre- and post-incidence matrices $\mathbf{C}^-$ and $\mathbf{C}^+$, respectively defined as follows:

1. if there exists an arc from z_i to t_{im} then $\mathbf{C}^-(z_i,t_{im})$="I_D", where I_D stands for "the function makes no transformation in the elements", otherwise $\mathbf{C}^-(z_i,t_{im})=0$.
2. if there exists an arc from t_{im} to z_m then $\mathbf{C}^+(z_m,t_{im})$="UP", where UP is a function that updates the colour <**rr**> with the colour <**rr**'>, otherwise $\mathbf{C}^+(z_m,t_{im})=0$. More precisely, **rr**' is the residual sequence of zones obtained from **rr** disregarding the first element z_i.

The set Ω is defined by Ω={Co(x): x∈$P\cup T$}. Moreover, considering that at the initial marking M_0 a route **r**(v) is assigned to each v∈V, M_0 is defined as follows: if $z_i\in P$ is the first zone of a route **r**(v) for some v∈V, then $M_0(z_i)$=<**r**(v)> else $M_0(z_i)$=<0>.

2.3 The AGVS dynamics

Now, to investigate the performance of the system it is convenient to extend the CPN with the time concept [6]. To do this we introduce a global clock, i.e., the clock values $\tau\in\Re^+$ represent the model continuous time, where $\Re^+$ is the set of non negative real numbers. Moreover, the temporisation of the Petri net is achieved by attaching a delay to the output arc of each transition, i.e., there is a delay after which the token becomes available. Here, the time delay $\delta(z_i,t_{qi})\in\Re^+$ is the time necessary for each AGV to move from zone z_q to zone z_i. Moreover, we allow each token to have a time stamp s(**rr**(v)) attached to it, in addition to the token colour **rr**(v). As soon as a token with colour **rr**(v)=<(z_i, z_m,…, z_1)> is in zone z_i, its stamp is reset to zero and the token will be available after $\delta(z_i,t_{qi})$ time units. Hence, transition t_{qi} is said to be *ready* for execution when the stamp s(**rr**(v)) satisfies condition s(**rr**(v))≥$\delta(z_i,t_{qi})$.

Moreover, transition t_{im} is enabled at marking M with respect to the colour <**rr**(v)>∈$Co(t_{im})$, if two conditions are simultaneously verified:

C1) $M(z_m)$=<0> if m≠1.

C2) $M(z_i)\geq\mathbf{C}^-(z_i,t_{im})$(<**rr**(v)>), with $M(z_i)$=<**rr**(v)> and **rr**(v)=(z_i z_m, …z_1).

In general, the firing of a transition t with respect to colour c∈Co(t) leads to a new marking M', denoted by M[t(c)>M'. A sequence $M[t_1(c_1)>M_1[t_2(c_2)>M_3\ldots M_{n-1}[t_n(c_n)>M'$ is denoted by M[ω>M', where $\omega=t_1(c_1)t_2(c_2)\ldots t_n(c_n)$ is a firing sequence: in this case we say that M' is reachable from M. Symbol R(M) denotes the set of reachable markings from M.

Example 1. Now, let us consider the AGVS described in subsection 2.1, with N_V=5 and the following routes assigned to the AGVs: $\mathbf{r}(v_1)=(z_7, z_2, z_3, z_8, z_7, z_1)$, $\mathbf{r}(v_2)=(z_2, z_3, z_8, z_7, z_1)$, $\mathbf{r}(v_3)=(z_9, z_8, z_{10}, z_1)$, $\mathbf{r}(v_4)=(z_3, z_8, z_7, z_1)$, $\mathbf{r}(v_5)=(z_1)$. We remark that v_5 is a vehicle waiting for destination in the docking station. Each time delay is $\delta(z_m,t_{im})=2$ for each $z_m\in P$ and $t_{im}\in T$. Figure 2 depicts the corresponding CTPN model at the initial marking M_0 and at time τ. The initial marking and the time stamps are defined as follows: $M_0(z_7)=<\mathbf{r}(v_1)>=<(z_7, z_2, z_3, z_8, z_7, z_1)>$, $M_0(z_2)=<\mathbf{r}(v_2)>=<(z_2, z_3, z_8, z_7, z_1)>$, $M_0(z_9)=<\mathbf{r}(v_3)>=<(z_9, z_8, z_7, z_1)>$, $M_0(z_3)=<\mathbf{r}(v_4)>=<(z_3, z_8, z_7, z_1)>$, $M_0(z_1)=<\mathbf{rr}(v_5)>=<(z_1)>$, $s(\mathbf{rr}(v_1))=0$, $s(\mathbf{rr}(v_2))=1$, $s(\mathbf{rr}(v_3))=3$, $s(\mathbf{rr}(v_4))=2$, $s(\mathbf{rr}(v_5))=5$. Figure 2 shows the colour and the associated with each token.

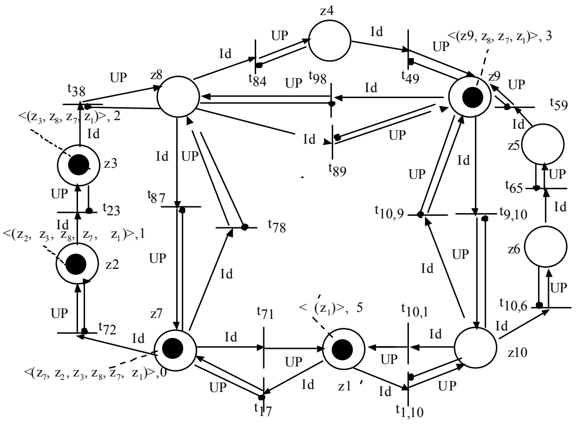

Figure 2. The CTPN at marking M_0 for the case study (N_V=5).

Two types of events can change the marking of the CTPN:

i) a new path $\mathbf{r}$ is assigned to a vehicle $v\in V$ (1-type event). This event is identified by the pair $\sigma_1=(v, \mathbf{r})$;
ii) a vehicle moves from a zone to another one (2-type event). This event is identified by the symbol σ_2=v.

The occurrence of a 1-type event $\sigma_1=(v, \mathbf{r})$ assigns the new colour $\mathbf{r}$ to v. On the other hand, if a 2-type event σ_2=v happens with $\mathbf{rr}(v)=(z_i, z_m, \ldots z_1)$, transition t_{im} fires in the CTPN and the marking M is updated. Let Σ_1 and Σ_2 respectively indicate the sets of 1- and 2- type events.

Now, in the AGVS behaviour some scheduling alternatives can be considered. For example, when several transitions are simultaneously colour enabled and ready at time τ, it is necessary to choose the AGV that has to move to the next zone first. To this aim, a priority rule is defined by a function associating with each marking M and each instant τ a vehicle from V, i.e., π: $M_A\times \overline{N} \rightarrow V$ where M_A is the set of admissible markings and $\overline{N}$ is the set of non-negative integers.

3 Previous results about deadlock conditions

Here, we recall the main definitions and results necessary to explain the deadlock avoidance strategies [4, 1]. The proposed techniques use digraphs to characterize deadlock: all the current interactions between vehicles and zones are described by means of a digraph $D_T(M)=(N,E_T(M))$ named *Transition digraph* and depending on the current CTPN marking. Each vertex in N corresponds to a zone z_i, so that the same symbol is used for vertices and zones, i.e., N=Z. The vertex set is fixed, but the edge set changes at each event occurrence and is defined as follows: $e_{im}\in E_T(M)$ iff there exists at marking M a vehicle $v\in V$ occupying z_i and requiring z_m as the next zone. The following is proven in [1]:

Proposition 1. The AGVS is in deadlock condition in the current marking M iff there exists a cycle in the transition digraph that does not contain zone z_1.

Now, a deadlock avoidance policy must prevent not only deadlock, but also unsafe states that are not deadlock but can incur a deadly embrace in the next future. To define an efficient deadlock avoidance strategy, a state named *second level deadlock* is characterized by the definition of a second digraph $D_R(M)=(N,E_R(M))$ called *residual path digraph* [1]. A digraph $D_R(M)$ is built taking into account the residual path that each vehicle in the system has to complete at the current marking M. Hence, $e_{im}\in E_R(M)$ iff z_m immediately follows z_i in $\mathbf{rr}(v)$, for some $v\in V$. Now, a second level deadlock can occur only if the cycles of $D_R(M)$ enjoy a particular property that can be exhibited using a further digraph, $D^2_R(M)=(N^2(M),E^2_R(M))$, named *second level digraph* and obtained from $D_R(M)$ as follows. Denoting by $\{\gamma_1, \gamma_2, \ldots, \gamma_M\}$ the complete set of the cycles of $D_R(M)$ not including z_1, we associate a vertex $\gamma_k\in N^2(M)$ to each cycle γ_k of $D_R(M)$. Moreover, an edge e^2_{kh} is in E^2_R iff the following two conditions hold: a)γ_h and γ_k have only one vertex in common (say r_m); b) there exists a residual path $\mathbf{rr}(v)$ for some $v\in V$ requiring zones z_i, z_m and z_p in strict order of succession and e_{im} is an edge of cycle γ_h, while e_{mp} is an edge of cycle γ_k.

Now, let γ^2 be a cycle from $D^2_R(M)$ (second level cycle) and let $\Gamma^2(M)$ be the subset of second level cycles enjoying the following property: $\gamma^2\in\Gamma^2(M)$ iff the cycles associated with the vertices of γ^2 are all disjoined but for one vertex, common to all of them. Moreover, let the capacity of a cycle γ (denoted by $C(\gamma)$) be defined as the number of resources involved in such a cycle. Analogously, let us define the capacity of a second level cycle $C(\gamma^2)$ as the number of distinct resources involved in all the cycles corresponding to the vertices of γ^2. Finally, let $C^2_0(M)$ be the minimum capacity of the second level cycles from $\Gamma^2(M)$ ($C^2_0(M)=\infty$ if $\Gamma^2(M)$ is empty). Considering that $n_V(M)$ indicates the number of vehicles performing transport operations in the current marking (not including the vehicles waiting in the docking station) the following proposition is proven in [1]:

Proposition 2: A marking M can be a second level deadlock state for the AGVS only if $\Gamma^2(M)$ is not empty and $n_V(M)\geq(C^2_0(M)-1)$.

We remind that a digraph containing N nodes is completely characterized by its (N×N) adjacency matrix [5]. Modelling the AGVS by the CTPN, it is possible to obtain the $(|P|\times|P|)$ adjacency matrices $\mathbf{A_T}(M)$ and $\mathbf{A_R}(M)$ of digraphs $D_T(M)$ and $D_R(M)$ respectively, by means of the incidence matrix of the CTPN at marking M [2, 3].

4 The control policies

This section recalls two control policies (CPs) to avoid deadlocks and collisions in zone-controlled AGVSs [1, 3]. The two policies share the same algorithm checking for *zone validation* and differ for the *path validation* algorithm.

4.1 Control policy 1

In terms of Petri nets modelling, a transition $t_{im}\in T$ is said to be controlled if its firing is determined by a CP when t_{im} is enabled according to conditions C1 and C2. Therefore, a CP is a mapping associating with each event $\sigma\in\Sigma_1\cup\Sigma_2$ and with each marking M a control action that enables and inhibits event σ, i.e., distinguishing between 1-type and 2-type events:
$f_i : \Sigma_i\times M_A\rightarrow\{0,1\}$ with i=1,2,
where $f_i(\sigma_i,M)=0$ ($f_i(\sigma_i,M)=1$) means that for the CTPN at marking M event σ_i is control inhibited (enabled) for i=1,2.

From the CTPN point of view, a deadlock means that once some marking has been reached, some colour enabled transitions cannot be fired any longer. The consequence in our model is that the travel of an AGV is interrupted and the vehicle can not reach the docking station. Accordingly, we define *task marking* M* the system state in which all the AGVs are in the docking station and have completed their assigned routes: $M^*(z_1)=N_V<z_1>$ (i.e., N_V tokens are in z_1 with colour $<z_1>$) and for each z_i such that $z_i\neq z_1$ it holds $M^*(z_i)=<0>$. Hence, a CTPN is deadlock-free at marking M under a CP iff there exists a controlled firing sequence ω so that $M'[\omega>M^*$.

Starting from *Propositions 1* and 2 a policy that is deadlock and restricted deadlock-free can be defined as follows [1].

CP1
Let the CTPN be at time τ and at marking M and denote with M' the marking obtained from M after the occurrence either of $\sigma_1\in\Sigma_1$ or of $\sigma_2\in\Sigma_2$.
$f_1(\sigma_1,M)=0$ if digraph $D_T(M')$ described by $\mathbf{A_T}(M')$ exhibits a cycle that does not include z_1 or if $\Gamma^2(M')$ is not empty and $n_V(M')\geq(C^2_0(M')-1)$.
$f_1(\sigma_1,M)=1$ elsewhere.
$f_2(\sigma_2,M)=0$ if digraph $D_T(M')$ described by $\mathbf{A_T}(M')$ exhibits a cycle that does not include z_1.
$f_2(\sigma_2,M)=1$ elsewhere.

The recalled control policy is the less restrictive policy proposed in [1], but its complexity can be high for AGVSs with a layout including a large number of bi-directional guide paths [1].

4.2 Control policy 2

To overcome the drawbacks of CP1, a second CP has been introduced based on the task marking reachability [3].

Definition 3: Let the CTPN be at marking $M'\in R(M_0)$ and at time τ, and let us suppose that $f_1(\sigma_1,M)=0$ for each $\sigma_1\in\Sigma_1$ and $M\in M_A$. The task marking M* is said to be reachable from M' under f_2 and the priority rule π, if there exists a controlled firing sequence ω so that $M'[\omega>M^*$.

Let the CTPN be at time τ and at marking M and let M' denote the marking obtained from M after the occurrence either of $\sigma_1\in\Sigma_1$ or of $\sigma_2\in\Sigma_2$. The CP is defined as follows.

CP2
$f_1(\sigma_1,M)=1$ if M*is reachable from M under f_2 and priority rule π.
$f_1(\sigma_1,M)=0$ elsewhere.
$f_2(\sigma_2,M)=0$ if the digraph $D_T(M')$ and described by $\mathbf{A_T}(M')$ exhibits a cycle that does not include z_1.
$f_2(\sigma_2,M)=1$ elsewhere.

It is proven in [3] that an AGVS under CP2 is deadlock and restricted deadlock free. Moreover, the check performed by f_2 works in polynomial time.

5 Synthesis of the controller

This section describes the activities of the real time controller: the Zone Validation and the Path Validation Algorithms (ZVA, PVA) that implement CP1 and CP2. While the ZVA is the same for CP1 and CP2, these differ for the PVA.

5.1 Zone validation

Let us suppose that the CTPN is at marking M and that transition t_{im} is colour enabled, i.e., $M(z_i)=<\mathbf{rr}(v)>=<(z_i, z_m, \ldots z_1)>$. If $s(v)\geq\delta(z_i)$ holds, event $\sigma_2=v$ can occur. In such a case, the controller executes the following Zone Validation Algorithm (ZVA).

ZVA
A1If $M(z_m)\neq<0>$ by C1) the transition can not fire, the zone is not validated. Go to step A5.
A2If $M(z_m)=<0>$ then the controller determines the new marking: $M'(z_i)=<0>$; $M'(z_m)=<z_m, \ldots z_1>$; $M'(z)=M(z)$ for each $z\in P$ with $z\neq z_m$, and $z\neq z_i$.
A3Build $\mathbf{A_T}(M')$.
A4A depth-first search algorithm [7] is applied to $\mathbf{A_T}(M')$: if the search finds a cycle not including z_1 then $f_2(\sigma_2,M)=0$ and go to step A5, else $f_2(\sigma_2,M)=1$ and go to step A6.
A5The zone is not validated, the behaviour of the AGVS continues with the CTPN at marking M. STOP.
A6The zone is validated and t_{im} fires, the behaviour of the AGVS continues with the CTPN at marking M'. STOP.

5.2 Path validation

Path validation consists in enabling or inhibiting 1-type events to avoid deadlocks and restricted deadlocks as a new path is assigned to some vehicle. Let **r** be the path that the scheduler proposes for vehicle v and let M be the current marking of the CTPN at time τ. Under CP1, the controller executes the following PVA.

PVA1
B1 Update the marking of the CTPN as follows: $M'(z_k)=<\mathbf{r}(v)>$ where z_k is the first resource of $\mathbf{r}(v)$ $M'(z_i)=M(z_i)$ for each $z_i\in P$ with $z_i\neq z_k$.
B2 A depth-first search algorithm is applied to $\mathbf{A_T}(M')$: if the search finds a cycle not including z_1 then $f_1(\sigma_1,M)=0$ and go to step B4.
B3 Build the path digraph and the second level digraph. If $\Gamma^2(M')$ is not empty and $n_V(M')\geq(C^2_0(M')-1)$, then $f_1(\sigma_1,M)=0$ and go to step B4, else go to step B5.
B4 $\mathbf{r}(v)$ is not validated and the evolution of the CTPN

continues starting at marking M. STOP.

B5 **r**(v) is validated and the evolution of the CTPN continues starting at marking M'. STOP.

Under CP2, the real time controller has to validate the proposed route by using the following PVA2.

PVA2

B1 Update the marking of the CTPN as follows: $M'(z_k)=<\mathbf{r}(v)>$ where z_k is the first resource of $\mathbf{r}(v)$ $M'(z_i)=M(z_i)$ for each $z_i \in P$ with $z_i \neq z_k$.

B2 Start a simulation run that under ZVA and a priority rule π checks that the CTPN reaches marking M* from M'.

B5 If marking M* is reached then **r**(v) is validated and the evolution of the CTPN continues starting at marking M', else **r**(v) is not validated and the evolution of the CTPN continues starting at marking M. STOP.

6. Simulation study

This section presents a simulation analysis of the case study employing in turn the control policies CP1 and CP2, in order to compare their effectiveness. We investigate, for equivalent settings, which CP restricts the system congestion while guaranteeing that the system vehicles the system reach their destination at an early time. In particular, the purpose is to simulate the CTPN while assigning the vehicles the longest and most varied paths. In the simulations the AGV fleet size is varied, in order to check the control policy effectiveness when traffic changes from low to medium and finally becomes intense.

The CTPN representing the system is implemented in the MATLAB-Stateflow environment [8], where it is possible to integrate modelling and simulation of Stateflow event-driven systems (e.g., the AGVS dynamics) with the execution of MATLAB computation routines (e.g., the ZVA and PVA algorithms), while keeping track of time by way of a software clock. More precisely, here the Petri net places are represented by a finite state automaton, with sub-states modelling the zone shift of the AGV currently under investigation. As already mentioned, the main difference in the implementation of the two CPs is in the PVA definition. In particular, in the AGVS controlled by CP1 a "PVA1" block tests the presence of a particular cycle in the second level digraph originating from the new marking. Instead, in the system controlled by CP2 a "PVA2" block performs a simulation of the AGVS in order to check the task marking reaching under the ZVA control and supposing that no 1-type event occurs.

6.1 Description of the simulation experiments

A set of 300 routes resulting in three replications of a set of 100 paths, all starting and terminating with the docking station z_1, is considered for the AGVS to accomplish. The maximum route length is equal to ten, i.e., each route proposed by the path scheduler consists of ten zones, all ending with z_1. The assigned set of 300 routes is considered in six experiments for both control policies with an AGV fleet size N_V varying from 2 to 7. In order to keep into account the time necessary for an AGV to travel from a zone to an adjacent one, i.e., the time to move along a guide-path or accomplish a load (unload) operation from (to) a workstation or the corresponding buffer, we set a time for a vehicle to accomplish such zone crossing. For sake of simplicity, we assume that any AGV can travel between any couple of adjacent zones in 200 time units, i.e., in a time interval equal to the 500th part of the total run time. Now, we define the priority rule π as follows:

$$\pi(M,\tau)=v_k=\min_h \{s(v_h) \mid s(v_h)\geq\delta(z_i) \text{ with } M(z_i)=<\mathbf{rr}(v_h)>\}.$$

More precisely, the priority rule selects, among the vehicles enabling ready transitions, the one with minimum time stamp. In order to compare experiments employing a different number of vehicles and a different restriction policy, a fixed run period of T=100,000 time units is considered.

6.2 Performance measures

The measure of performance of the simulation experiments is the number of routes completed by the AGV fleet under the ZVA and PVA verifications in the run time T (throughput). Clearly, the number of completed missions is an index of the traffic management policy effectiveness in having the system accomplish paths while minimizing blockings. Note that only completed missions are considered in the throughput computation.

In the simulations vehicles utilization is evaluated. In particular, vehicles activity can be classified as follows.

(i) Loaded and booked travel – the vehicle is accomplishing a mission. This state comprises two sub-states: the AGV is either booked and travelling empty toward the loading zone, or transporting a piece toward the assigned unloading area.

(ii) Blocked – the AGV, travelling loaded or booked, is blocked in a zone by the control system to avoid deadlock or collision.

(iii) Empty and available – the vehicle, after unloading a part at the destination zone, is either empty and travelling towards the docking station or idle in the latter waiting for destination. At both stages the AGV can receive the next transportation task by the path scheduler. The measure of performance adopted for vehicles activity is the average percentage time a vehicle spends in each of the above three states. Additionally, we consider vehicles utilization, i.e., the average percentage time a vehicle spends outside the docking station.

6.3 Simulation results

The AGVS throughput is depicted in figure 3 for CP1 and CP2, respectively, when using different AGV fleet sizes. Figure 3 shows that the AGVS throughput is maximized under CP1 when five vehicles are available, whereas the highest number of accomplished transportation tasks is obtained under CP2 with six vehicles. Moreover, Figure 3 shows that the transportation system under both PVA strategies is not efficient for a modest AGV fleet size, since the number of vehicles is small compared with the AMS size as well as with the number of tasks to be completed. All the same, under both real time control strategies the AGVS is inefficient for a large AGV fleet size: traffic becomes congested and vehicles are often blocked in order to avoid deadlock and collisions. The average percentage time of loaded and booked travel under both CPs is reported in figure 4. Under both techniques the index decreases with the increase in AGV fleet size: this is expected, because of the corresponding traffic congestion. Figure 5 depicts the average percentage blocked time of vehicles. For both techniques the performance index increases with the fleet size. Indeed, the probability that a vehicle is blocked while executing a transportation task increases when the system is overcrowded. The average empty and available time is reported in figure 6. Such an index is only partly dependant on the adopted control policy: in fact, it is influenced by the selected dispatch policy and by the transport request rate, i.e., by the path scheduler design. Finally, the AGVS utilization index is reported in figure 7. Here, the average percentage time a vehicle is either booked travelling toward a loading station, or loaded and blocked, or else carrying a piece towards its destination is reported. This index is high for both CPs: most of the time vehicles are busy out of the docking station. We observe that CP2 outperforms CP1 in all the performance indices reported in figures 3-7. In fact, the path validation algorithm based on the task marking reaching tends to reject a lower number of tasks with respect to the algorithm based on cycles detection in the digraphs associated to the AGVS.

7. Conclusions

This paper presents a performance based comparison of two control policies (CPs) previously presented by one of the authors for zone controlled Automated Guided Vehicle Systems (AGVSs). The two CPs share the same algorithm checking for zone validation (ZVA) and differ for the path validation algorithm (PVA). A case study is considered. The AGVS structure and dynamics are described by a resource oriented Coloured Timed Petri Net (CTPN), allowing the investigation of the system performance. We compare the two CPs on the basis of appropriate performance indices in the MATLAB-Stateflow environment. Simulation evidence show the effectiveness of one of the considered control strategies compared to the alternative policy.

References

[1] Fanti, M. P., Event-based controller to avoid deadlock and collisions in zone control AGVS. *International Journal of Production Research*, **40**(6), 1453-1478, 2002.

[2] Fanti, M. P., Deadlock-Free Control in Automated Guided Vehicle Systems. In P. Ezhilchelvan and A. Romanovsky (eds.), *Concurrency in Dependable Computing*, The Netherlands: Kluwer Academic Publishers, 2002.

[3] Fanti, M. P., A Deadlock Avoidance Strategy for AGV Systems Modelled by Coloured Petri Nets, *Proceedings of the 6th Workshop on Discrete Events Systems*, Zaragoza, Spain, 2002.

[4] Fanti, M. P., Maione, B., Mascolo, S., and Turchiano, B., Event Based Feedback Control for Deadlock Avoidance in Flexible Production Systems. *IEEE Transactions on Robotics and Automation*, **13**(3), 347-363, 1997.

[5] Harary, F., *Graph Theory*, Reading: Addison-Wesley Publishing Company, 1971.

[6] Jensen, K., *Colored Petri Nets, Basic Concepts, Analysis methods and Practical Use*, Vol I, EATS Monography and Theoretical Computer Science, New York: Springer Verlag, 1992.

[7] Reingold, E. M., Nievergelt, J., and Deo, N., *Combinatorial Algorithms. Theory and* Practice, New Jersey: Prentice-Hall, 1977.

[8] The MathWorks Inc., *Stateflow User's Guide Version 4*, 1997.

[9] Viswanadham, N., and Narahari, Y., *Performance Modeling of Automated Manufacturing* Systems, Englewood Cliff, NJ: Prentice Hall, 1992.

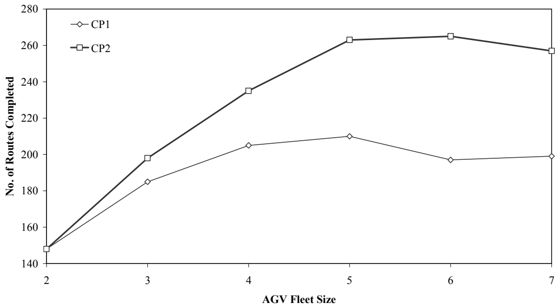

Figure 3. Throughput for various AGV fleet sizes.

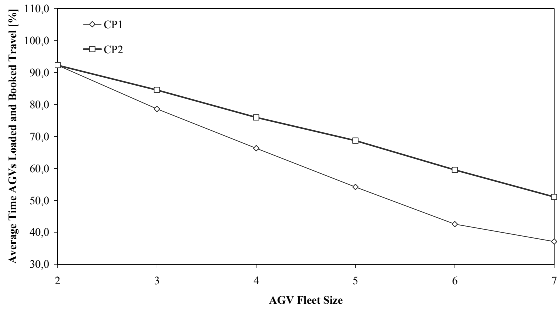

Figure 4. Average time vehicle loaded and booked travel.

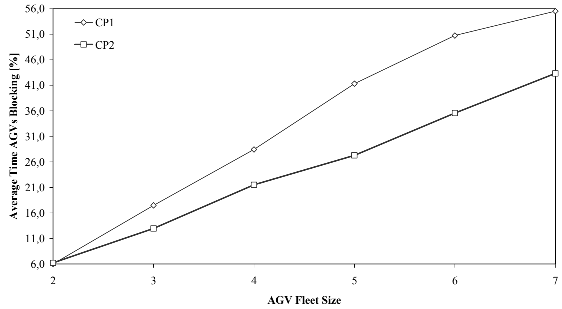

Figure 5. Average time vehicle blocking.

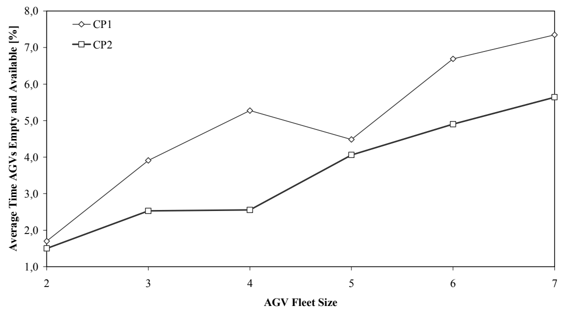

Figure 6. Average time vehicle empty and available.

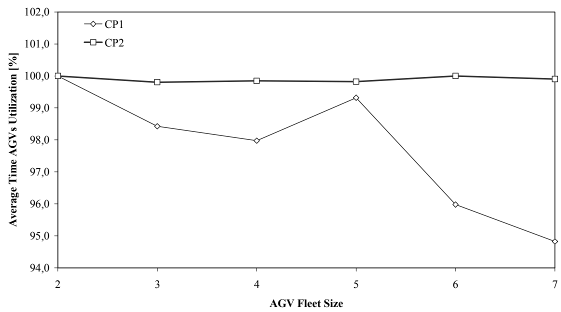

Figure 7. Average time vehicle utilization.