STOCHASTIC AND COLOURED PETRI NETS FOR MODELLING AND DESIGNING OF AGV SYSTEMS*

Rui Paulo Rocha and Adriano Silva Carvalho

Electrical Engineering and Computers Department & Institute for Systems and Robotics
Faculty of Engineering, University of Porto, Portugal
E-mails: {rpr, asc}@fe.up.pt

ABSTRACT: Nowadays the materials flow in manufacturing systems is often assured by AGVs. This paper describes a typical AGV System and presents a three layers control architecture. Petri nets are a well-known modelling tool for Discrete Event Dynamic Systems. They are widely used in simulation of systems from different application areas, which exhibit concurrency, synchronisation and resources sharing. Within this context, the application of Stochastic and Coloured Petri nets to the modelling and designing of AGVs Systems is described. After establishing a Stochastic and Coloured Petri net model, a case study for analysis of management and control strategies for an AGVs fleet is presented.

INTRODUCTION

Currently, companies are using flexible manufacturing systems, as a way to improve their technical modernisation. With the increasing globalisation of the competitive manufacturing arena and the resulting shift from economies of scale to economies of scope, many companies have to adjust to rapidly changing market conditions, while cutting down their production costs in order to survive [1].

One way to achieve this is to give careful consideration to strategic and logistical issues, like facility layout, production management, material flow automation, supply management and inventory control, at early stages of design of manufacturing environments. Related design questions often can be transcribed into the form of large-scale optimisation problems with tight constraints, and requiring accurate models and precise solutions.

Most of the technical evolution related with the materials flow automation in manufacturing systems, have centred at low levels, very closed to hardware, resulting in the appearance of more sophisticated machines, with minors processing and setup times. As a consequence of this evolution, the production planning tends to give little importance to the execution time of tasks, being worth the efficiency with which materials are carried between workstations or between these ones and warehouses [1-2]. In

* This work has been possible thanks to the PhD subvention PRAXIS XXI / BD / 18518 / 98, granted by the Ministry of the Science and the Technology of Portugal, and thanks to co-operation between the University of Porto and the Portuguese Company EFACEC Automação e Robótica.

material handling systems, based on AGVs. Hereafter they will be referred as AGV Systems (AGVSs).

Excluding continuous process industries, manufacturing systems belong to the class of Discrete Event Dynamic Systems (DEDSs), whose behaviour is defined by time evolution synchronised by occurrence of events, that involves abrupt state transitions (e.g. end of machining time, new batch arrival, equipment failure, etc.) [3].

Despite research efforts applied to the development of new mathematical methods for analysis of this type of system (e.g. Markov chains, queuing theory, etc.), the use of analytical models is possible only supported on very simplified models, whose validity must be verified. If the objective is to choose a design strategy from a set of strategies, usually the analytical models serve only to analyse the system qualitatively, eliminating firstly the bad options. However, it does not allow to choose the best one [4].

Without modelling and quantitative evaluation techniques, it becomes a hard task to predict accurately the behaviour of a real manufacturing system. It adds that it is usual to take into consideration stochastic information of the system, like processing times, failures and repairs, with related effect on performance figures. So a lot of techniques for modelling and quantitative analysis of DEDSs have been studied and carried out to overcome this problem. Among them, Petri nets are now considered one of the most powerful paradigms, especially suitable for systems that exhibit concurrency, conflicts and synchronisation.

This paper describes the use of Stochastic and Coloured Petri nets (SCPNs) as a modelling and designing tool for an AGVS integrated into a manufacturing system [5]. Within this context, an AGVS is briefly described and a control architecture for a typical wire guided AGVS is established. Using the Artifex software package for graphical editing and animated simulation of SCPNs, an accurate and complete SCPN model for a generic wire guided AGVS has been built.

The SCPN model has been configured for an AGVS case study, which allows to search for main design issues, affecting the performance of an AGVS. A comparative study about several alternative management strategies has been performed. Statistical analysis of the results and main conclusions of the study are finally discussed.

CHARACTERISATION OF A TYPICAL AGV SYSTEM

An AGV (Automatic Guided Vehicle) is a mobile device for automatic handling and transportation of materials. Its main functional features are:

- to have autonomy to travel within the layout of a manufacturing facility without human intervention;
- to be programmed to select paths and to stop in pickup and deposit stations with accuracy.

AGVs are considered the most flexible type of material handling systems. These vehicles, which range from mail deliverers to 125 ton transporters, are equipped with a variety of functions, such as guidance and routing systems to traffic management and load transfers. While operating within a manufacturing or service environment, the AGVs require conditions on-board or centralised computer control to co-ordinate the movements in respect to other material handling devices or other AGVs. Manual or automatic load transfer may be achieved once an AGV has reached its destination. An operator can board the AGV and position the vehicle to be loaded or unloaded. Otherwise, the AGV can transfer loads automatically using roller conveyors, clamping devices, lift tops, etc., depending on the material requirements.

Several methods of guidance can be implemented, which allow the AGV to follow a fixed path or a free path. The determination of fixed or free paths will depend on cost, flexibility requirements, and future expansion alternatives. While fixed path systems may be less expensive, the installation may disrupt production, causing subsequent costs. The wire guidance is an example of a fixed path method. Free path guidance requires a software programmable path and is more flexible to layout changes and capacity requirements than a fixed path layout. GPS guidance, laser guidance and artificial vision guidance are examples of free path methods.

Due to its robustness and simplicity, the wire guidance is the most common navigation system, where an implantation of loops embedded in the floor of the manufacturing facility layout, traversed by sinusoidal alternate electrical currents of several frequencies, generates equally sinusoidal magnetic fields. In this type of system, the AGV is equipped with a magnetic sensor, consisting of a pair of coupled solenoids in a differential assembly. An AGV travels along a path, following the magnetic field of each one of the layout sections that define it, characterised by a determinate frequency which is different from the magnetic field frequencies of the neighbour sections [5]. Figure 1 shows two of the wire-guided AGVs installed in the industrial transformers factory of the company EFACEC in Portugal.

Figure 1. AGVs in the industrial transformers factory of the company EFACEC in Portugal.

Usually, an AGVS comprises a fleet of some AGVs, controlled by a central computer. The AGVS controller must solve in real time the problems of:

- Scheduling - assignment of the AGVs to the pending movement tasks waiting in the queue, using a set of optimisation criteria [6];
- Routing - execution of a routing algorithm, which uses some optimisation criteria (e.g. minimum travel time) to determine the best path for an AGV, when it goes from its initial position until a pick up point, or when it goes from a pick on point to a drop off point [2];
- Traffic Managing - management of the layout physical space [5], preventing the collision among AGVs and resolving eventual conflicts or situations of deadlock [7];
- Dispatching and Monitoring of AGVs state - controlling and execution of predefined scheduling and monitoring of the AGVs state (e.g. diagnosing of AGVs failures) [5];
- Interfacing with the shop floor controller - reception of new movement orders requested by the shop floor controller, state monitoring of the pending orders, confirmation of movements and synchronisation with peripherals (e.g. roller conveyor) in picking up and dropping off operations [5].

CONTROL ARCHITECTURE OF AN AGV SYSTEM

The previous section has identified the main issues related with the management and control of AGVS. If we analyse the real-time requirements and the granularity of each control task, it is possible to classify them in three groups:

- Critical real-time control of each individual AGV in the fleet, namely control of execution of simple movements (movement between two layout nodes and pick up or drop off operation), synchronisation with other devices in the plant during pick up or drop off operations, monitoring of the AGVs state (e.g. low-battery alarm, hardware error, etc.) and reaction to some events resultant from the interaction with external environment (e.g. press-button actuation, detection of obstacles during a movement, etc.);
- Critical real-time control of the entire AGVs fleet, which is related with some decisions affecting all the AGVs in the fleet, namely scheduling, routing and traffic management;
- Real-time control of the interaction between the AGVS and the plant controller, namely reception of new movement orders, cancellation of pending orders, monitoring of pending orders and AGVS setup (e.g. number of enabled AGVs).

This classification induces a control architecture for AGVS with three layers. Figure 2 shows the block diagram of such architecture [5]. The block PLANT represents the manufacturing system where the AGVS is integrated, while the other three blocks represent the control layers. The arrows indicate different data flows that may exist among the 4 blocks of the system. The most common messages exchanged between each pair of blocks are expressed near to respective arrows.

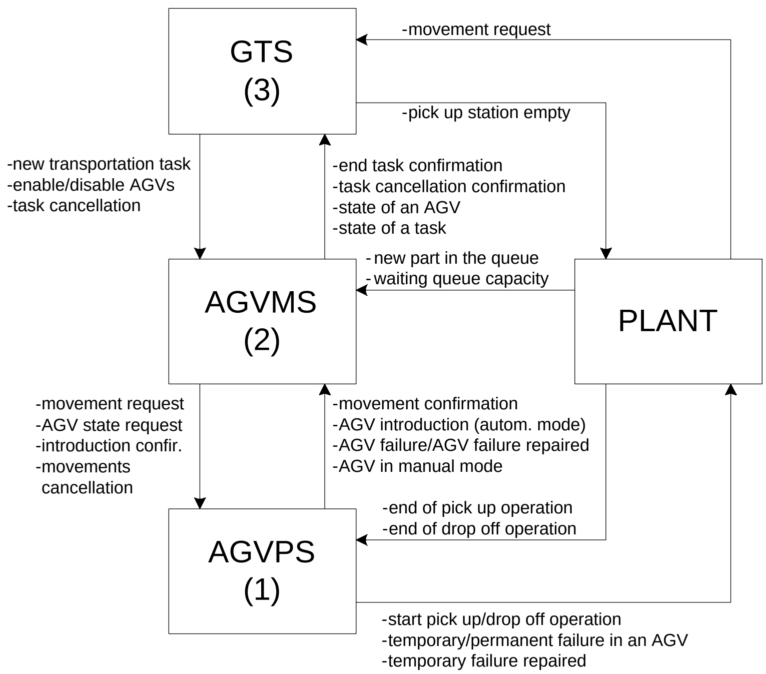

Figure 2. Control architecture of an AGVS.

The 3 layers of the control architecture are:

1. AGVPS - AGVs Positioning System - layer that interacts directly with the AGVs. The orders related to elementary movements proceed from the upper layer and are mapped in instructions of the AGVs programming language. This layer receives data about the state of the AGVs (e.g. failure diagnostics), managing the operation of AGVs in the system (transition of AGVs between the manual mode and the automatic mode). Namely it

synchronises the AGV with other devices in the plant, acting in a pick up or in a drop off operation (e.g. transfer table, roller conveyor, etc.).

2. AGVMS - AGVs Management System - it is the control layer with bigger computational weight. It carries through the functions of scheduling of movement tasks, routing the AGVs assigned to movement tasks, managing the traffic and resolving conflicts. When the scheduling strategy presupposes the knowledge of the length of the waiting queues in the pick up stations, it becomes necessary the data flow PLANT→AGVMS, through which this information is addressed.
3. GTS - General Transportation System - establishes the interface between the AGVS and the shop floor controller. It sends transportation requests and pending tasks cancellation requests to the inferior layer; it tracks the state of the AGVs and the pending tasks; it performs the setup of the fleet of AGVs, enabling and disabling individually each AGV. It has autonomy to reserve AGVs for the local scheduling of urgent or maintenance movement tasks (bypass to the AGVMS scheduler).

STOCHASTIC AND COLOURED PETRI NETS

Petri nets are a graphical modelling formalism, which may represent concurrent and co-operative processes, as well as relations existing between conditions and events [8]. They constitute a rigorous and powerful formalism for the modelling and analysis of DEDSs (Discrete Event Dynamic Systems), like discrete parts manufacturing systems. Structurally, they facilitate the graphical representation of behaviours, such as: concurrency, synchronisation, resources sharing, etc. [5].

Moreover, the structure of a Petri net may be described mathematically through an algebraic model. Thanks to this important feature, systematic analysis techniques of the Petri nets properties have been developed, such as: necessary and sufficient conditions for the verification of determined structural properties [9]; coverability tree and graph of markings; reduction methods; computation of marking and firing invariants; etc. [8].

In an autonomous Petri net, an enabled transition may fire, but it is not known when it is really fired. This type of model is useful to analyse structural properties of the systems being modelled, but it does not allow evaluating the performance of the same ones. Therefore, it was necessary to enrich the autonomous Petri nets formalism, synchronising the firing of the transitions with external events and/or introducing time. Thus, it appeared the concept of non-autonomous Petri nets [8].

Some examples of non-autonomous Petri nets are:

- Synchronised and interpreted Petri nets - they allow to synchronise the firing of a transition with the occurrence of an external event, as well as associating with the firing of the transition the execution of a specified action [8];
- Timed Petri nets - they allow to associate deterministic times to the places (P-timed Petri nets) or to the transitions (T-timed Petri nets); in the P-timed Petri nets, after a token is deposited in a place, that one has to remain there during a minimum specified time, before it is consumed by a transition firing; in the T-timed Petri nets, it is specified the time which mediates between the enabling and the firing of a transition [8];
- Stochastic Petri nets (SPNs) - extension of the T-timed Petri nets that allows to associate stochastic firing times with transitions, according to a probability density function (PDF). In its initial definition, all transitions have firing times that follow an exponential PDF [10]. The Generalised SPNs [11] and the Deterministic SPNs [12] allow to extend the initial model of the SPN, in order to support transitions with deterministic firing times.

With the increase of the interpretation level of the Petri net models, they have become useful to evaluate the performance of DEDSs). In order to make possible the use of Petri nets for modelling more complex systems, it was also necessary an evolution of the paradigm in the way of increasing the abstraction level of the models. Thus, it has appeared the high level Petri nets, being to detach in this context the definition of the Coloured Petri nets formalism (CPNs) by Kurt Jensen [13]. With the CPNs, the models had become more compact, because it became possible to define and to manipulate tokens with attributes (colours) and thus to distinguish the tokens between themselves. Moreover, the CPNs has introduced the modularity concept, which allows to give a more consistent structure to the models and to partition them into sub-nets, which make the models more intelligible.

Nowadays, the Stochastic and Coloured Petri nets (SCPNs) are considered an extremely useful formalism to analyse and to evaluate the performance of complex systems, where the times associated to the occurrence of the events are random variates (e.g. discrete part manufacturing systems).

A SCPN MODEL FOR AN AGV SYSTEM

The architecture of Figure 2 has been the basis for the construction of a SCPN model for an AGVS [5]. The modelling software environment used to build the model has been the Artifex Software Package, which supports:

- graphical edition of SCPN models, being the transition predicates and the transition actions written in C programming language;
- animated execution and validation of the models;
- recording and visualisation of the data collected during a simulation, as well as its exportation for formats compatible with other statistical analysis software applications (e.g. Excel).

The Artifex allows the definition of sub-nets, partitioning the models and making them more intelligible. Moreover, it makes possible the definition of object classes, that are SCPNs eventually defined by instances of other object classes and the interconnection of some SCPN sub-nets. An object possesses a set of output and input places, used to communicate with other objects. Being defined an object class, it is possible to create instances of objects of that class in another model of higher level, allowing the incremental construction of models and the creation of objects libraries, which may be reused in the future, as necessary.

Figure 3 shows the high-level view of the Artifex model for an AGVS, with the architecture represented in Figure 2. It consists of 4 objects: 3 objects representing the 3 control layers of the AGVS; and one object representing the manufacturing system plant.

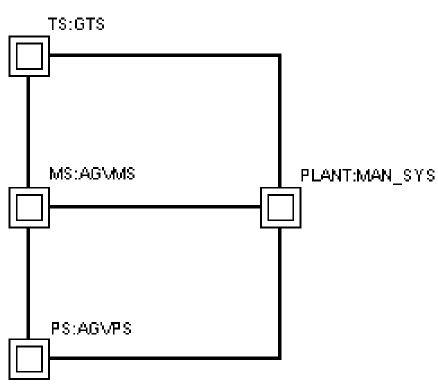

Figure 3. High-level view of the Artifex (SCPN) model for an AGVS.

The main entities of an AGVS - the AGV, the Task and the Layout - have been modelled in such a way that it makes possible the configuration of the more relevant system parameters and the customisation of the model to a specific AGVS. The interconnections among the 4 objects simulate the messages exchanging defined in the architecture of Figure 2.

STUDY ABOUT AGVS MANAGEMENT STRATEGIES

The model shown in Figure 3 has been configured for the flexible manufacturing system represented in Figure 4 , used as a case study for AGVs management strategies [5].

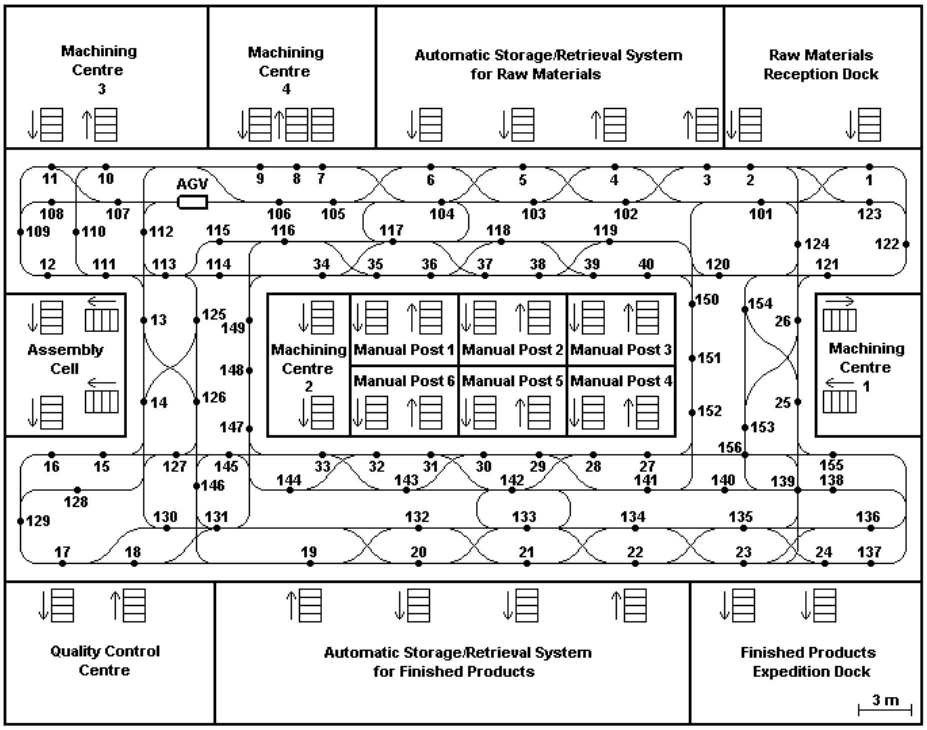

Figure 4. Flexible manufacturing system used to study AGVs management strategies.

For the same physical parameters of the AGVs and the plant (AGVs speeds, AGVs failure rates, technological paths, queues capacities, warehouses configuration, etc.), and modifying only the AGVs management strategy, 8 simulations of the system have been performed, throughout one day of work.

Table 1 shows the management strategies used in each simulation [5]. Each management strategy is the combination of two rules to be used in:

- AGV initiated assignment - used in a scenario with more tasks than AGVs; when an AGV finishes the execution of a task, it is chosen a new task for the available AGV;
- Work centre initiated assignment - used in a scenario with more AGVs than tasks; when a work centre requests a new movement task, it is chosen one of the available AGVs to execute the new task.

TABLE 1
Management Strategies Used in Each Simulation.

Sim. Nr.	AGV initiated assignment rule	Work centre initiated assignment rule
1	Shortest Travel Time (STT)	Nearest Vehicle (NV)
2	Maximum Outgoing Queue Size (MOQS)	
3	Minimum Remaining Outgoing Queue Space (MROQS)	
4	Modified First Come-First Serve (MFCFS)	
5	Shortest Travel Time (STT)	Least Utilised Vehicle (LUV)
6	Maximum Outgoing Queue Size (MOQS)	
7	Minimum Remaining Outgoing Queue Space (MROQS)	
8	Modified First Come-First Serve (MFCFS)	

Figure 5 shows a graph that allows comparing the average of the total operation time and the total execution time of a task, as a function of the management strategies.

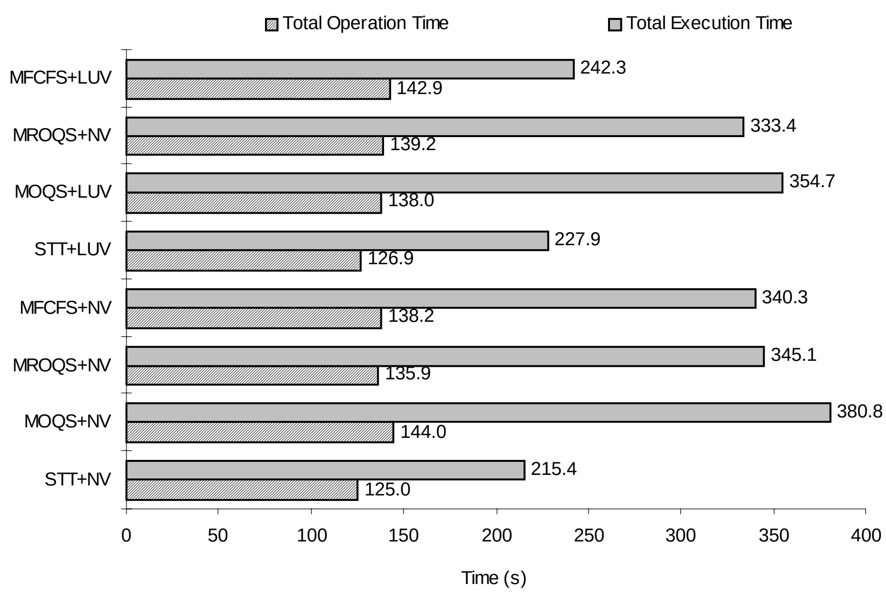

Figure 5. Average of the total operation time and the total execution time of a task.

The total operation time of a task (t_{ot}) is the sum of the following partial times:

- travel time of the AGV since its initial location until the pick up station (t_1);
- load time of the item to carry (t_2);
- travel time of the AGV since the pick up station until the drop off station (t_3);
- drop off time of the carried item (t_4).

The total execution time of a task (t_{et}) is the sum of the total operation time with the waiting time of the task in the queue (t_q).

The management strategies rule t_{et} through parcels t_1 and t_q, because some strategies choose the AGV with lower t_1, while others use another type of criteria (e.g. waiting queue length in the pick up points), which prejudices t_1 and optimises t_q. Obviously, the time t_{ot} suffers the influence of the strategy choice through the parcel t_1.

Observing time t_{ot} in the graph of Figure 5, we can conclude that time t_1 presents a low variation along the simulations. The main differences in system performance are observed in the time t_{et}. This evidence allows concluding that the performance of the AGVS is more sensible to the average waiting time of the tasks in the queue (t_q) than to the time t_1.

As it was discussed, the AGV initiated assignment rule influences much more the AGVS performance than the work centre initiated assignment rule, because it is much more frequent to have tasks competing for an available AGV than the inverse situation. Notice, for example, the small difference between the results of simulations 1 (STT+NV) and 5 (STT+LUV).

The graph of Figure 5 allows to choose the combination STT+NV as the best combination of management strategies, followed closely by the combination STT+LUV, because of the reason previously explained.

Combination MFCFS+NV appears in third place, with a good optimisation of the time t_q, but with one of the worst results in time t_1. Because of the first come-first serve policy, the combination allows the best optimisation of the time t_q. However, optimisation in time t_1 is very poor, when compared with the STT strategy.

The use of the strategies MOQS and MROQS allows balancing the length of the waiting queues in the drop off stations in a more efficient way than the other strategies. This allows the system to have a more stable operation, preventing the appearance of deadlocks related with the saturation of the waiting queues. Although these strategies improve the transitory behaviour of AGVS, the times t_{ot} and t_{et} are better optimised by other strategies.

CONCLUSIONS

AGV Systems (AGVSs) are used in the current flexible manufacturing systems, as a way to achieve high levels of material flow optimisation. Due to their complexity and discrete event nature, they are often studied through simulation models, because the computation of analytical solutions for mathematical discrete event models of real systems is usually not feasible. Petri nets constitute one of the most consistent modelling formalisms for Discrete Event Dynamic Systems (DEDSs). Thanks to the increase of interpretation and abstraction levels of models, the use of Stochastic and Coloured Petri nets (SCPNs) in analysis and performance evaluation of DEDSs presents innumerable advantages.

This paper has identified the main control issues of typical AGVS and it has defined a control and management architecture of three layers. Based on this architecture a detailed and generic SCPN model of an AGVS has been built. This model has been customised for a case study of a flexible manufacturing system, which is the basis for a study of AGVs management strategies. They are applied some strategies optimising the travel time between the initial location of AGV and the pick up point of the task and some strategies optimising the waiting time in the queue.

The result analysis of the study has focused at total time of operation and at total time of execution of a movement task. The first one does not include the waiting time in the queue, while the other one includes it. The total time of execution is the more critical performance parameter of the AGVS, due to waiting time of a task in queue. This time has revealed a much more critical parameter than the travel time of the AGV from its initial location to the pick up station.

The strategies based on the length of the waiting queues in the pick up stations allow a stable operation, preventing the appearance of deadlocks related with the saturation of the queues.

Although these strategies improve the transitory behaviour of the AGVS, they are not the best choice from a steady-state point of view. The combination Short Travel Time rule and Nearest Vehicle rule is the best control strategy among the eight combinations intervening in the study, as it allows the best throughput for the system. This rules combination optimises the travel time between the initial location of AGV and the pick up point.

The authors are currently extending the presented study to the designing and modelling of other material handling systems, through the use of Object Oriented Petri nets which are currently being developed by many people of the Petri nets area. The new Petri nets extensions will lead to a bigger abstraction level and a more effective reusing capacity of the models.

REFERENCES

[1] **Reembold, U, Nuagi, B. O., Store, A.**, *Computer Integrated Manufacturing and Engineering*, Addison and Wesley Publishing Company, 1995.

[2] **Taghaboni-Dutta, F., Tanchoco, J. M. A.**, *Comparison of Dynamic Routing Techniques for Automatic Guided Vehicle Systems*, International Journal of Production Research, Vol. 33, Nr. 10, pp. 2653-2669, 1995.

[3] **Cassandras, C. G.**, *Discrete Event Systems: Modeling and Performance Analysis*, Aksen Associates Incorporated Publishers, 1993.

[4] **Buzacott, J. A., Shanthikumar, J., G.**, *Stochastic Models of Manufacturing Systems*, Prentice-Hall, 1993.

[5] **Rocha, R. P., Development of an AGVs Management System**, MSc Dissertation, Faculty of Engineering, University of Porto, Set. 1998.

[6] **Eglebu, P. J., Tanchoco, J. M. A.**, *Characterisation of Automatic Guided Vehicle Dispatching Rules*, International Journal of Production Research, Vol. 22, Nr. 3, pp. 359-374, 1984.

[7] **Kim, C. W., Tanchoco, J. M. A., Koo, P.**, *Deadlock Prevention in Manufacturing Systems with AGV Systems: Banker's Algorithm Approach*, Journal of Manufacturing Science and Engineering, Vol. 119, pp. 849-854, 1997.

[8] **David, R., Alla, H.**, *Petri Nets and Grafcet - Tools for Modelling Discrete Event*, Prentice-Hall, 1992.

[9] **Murata, T.**, *Petri Nets: Properties, Analysis and Applications*, Proceedings of the IEEE, Vol. 77, Nr. 4, Abr. 1989.

[10] **Molloy, M. K.**, *Performance Analysis using Stochastic Petri Nets*, IEEE Transactions on Computers, Vol. 31, pp. 913-917, 1982.

[11] **Marsan, M. A., Balbo, G., Conte G.**, *A Class of Generalised Stochastic Petri Nets for the Performance Analysis of Multiprocessor Systems*, ACM Transactions on Computer Systems, pp. 93-122, 1984

[12] **Marsan, M. A., Chiola, G.**, *On Petri Nets with Deterministic and Exponentially Distributed Firing Times*, Advances in Petri Nets, Lecture Notes in Computer Science, Springer, Vol. 266, pp. 132-145, 1987.

[13] **Jensen, K.**, *A Brief Introduction to Coloured Petri Nets*, Computer Science Department, University of Aarhus, Denmark, 1997.