Supervisory Control of AGV's for Flexible Manufacturing Cells

J. C. Pérez-Sampieri*, E. Aranda-Bricaire*, E. G. Hernández-Martínez**
**Mechatronics Section, CINVESTAV, AP 14-740,*
7000 Mexico D.F., (e-mail: rakbreak@gmail.com, earanda@cinvestav.mx)
***Engineering Department, Universidad Iberoamericana, 01219 Mexico D.F.,*
(e-mail: eduardo.gamaliel@ibero.mx)

***Abstract*— This paper presents a motion coordination strategy based on a hierarchical control architecture for a flexible manufacturing cell equipped with automated guided vehicles (AGVs). The AGV's transport raw material among different workstations and automated warehouses. The hierarchical control architecture is divided into two levels. The high level includes a Discrete-Event plant model using the Finite-State Automata formalism and two supervisors are synthesized to enable concurrent tasks obeying process restrictions, and the product sequences, respectively. In the low level, the transportation tasks are translated to each AGV using motion coordination control laws based on artificial vector fields to guarantee convergence to the goals. Repulsive vector fields are also employed to avoid inter-robot collisions. The approach was tested in both virtual reality environment and a experimental setup with two AGV's for a specific product sequence.**

Keywords: Supervisory Control, Mobile Robots, Flexible Manufacturing Cell, Unicycle-type robots.

I. INTRODUCTION

Multiple and concurrent tasks are enabled in Flexible Manufacturing Cells (FMC) to create several products in order to fulfill the highly demanding markets (Groover, 2007). The modeling and control of FMC have been widely studied by the DES community using the Finite State Automata (FSA) or Petri Nets (Cassandras and Lafortune, 2008). The FSA approach allows to construct complex plant model through the synchronous product of partial models. It also allows to synthesize supervisors that satisfy process restrictions like storage limitations, task precedence, shared resources, etc.

The material-handling system of a FMC is crucial to increase its productivity. It is commonly composed by robots manipulators, gantries over rail or conveyor belts that transport raw material, sub-parts and finished products between workstations and automatic warehouses. Recently, the industry has focused on a more intelligent FMC where a group of Automated Guided Vehicles (AGV's) replace the fixed material-handling system to achieve a flexible transportation environment (Cao et al., 1997). The AGV's coordination implies a multidisciplinary field of study where it is required to design decentralized control laws for the point convergence, formations, collision avoidance, traffic control, etc. as exposed in (Hernández-Martínez, 2009).

The challenge of the AGV coordination in FMC involves the combination of a DES model of the FMC with the low level motion control laws in hybrid control architectures. Some related works are (Herrero-Pérez and Martínez-Barbera, 2010) where a Petri Net model solves problems of task allocation and traffic control, Directed Acyclic Graph (DAG) method is used to coordinate AGV's in multiple workstations with delivery conditions (Kim and Oron, 2012). The relational database (blackboard system) of the AGV's for materials handling, factory scheduling and transport systems is studied in (Farahvash and Boucher, 2004). A hierarchical decomposition of the information of the AGV's is used in (Makris et al., 2012) to define multiple targets and random motion sequences of the robots. All these previous works address only the conceptual ordering and task assignment of the AGV's in the FMC tasks, however a clear approach about the motion control laws of the AGV's has not been widely explored.

The main contribution of this paper is the combination of the coordination of AGV's based on continuous control laws and its interaction with a discrete-event model of a FMC obtained from industrial standards. The FMC studied is the classical configuration of an assembly FMC composed by a set of raw material warehouses, sub-part machines, assembly units and sub-parts and final products warehouses. The control is decomposed in two levels. In the high level, a FSA plant model and a supervisor is designed to obey some process restrictions. The task-based DES modeling uses the main ideas of the industrial standard ISA-95 (ISA, 2005), that proposes to model generic tasks according to the capacities of the equipment only, regardless of the product sequences. The DES supervisor identifies and organizes the transportation task realized by the AGV's. On the other hand, the AGV's are modeled as mobile wheeled unicycle-type robots, as defined in (González-Sierra et al., 2011; Canudas-de-Wit et al., 2012), where continuous control laws are designed to achieve specific positions with collision avoidance in the FMC. The previous hybrid architecture enables the concurrent behavior of the AGV's to achieve organized and optimized time production in the FMC.

The problem statement is discussed in Section II. Some preliminaries about the supervisory control theory and point convergence of the AGV's are presented in Section III. The hybrid architecture is given in Section IV. Experimental work with two AGV's in a product routine is explained in section V. Some conclusion remarks are offered in the Section VI.

II. PROBLEM STATEMENT

Figure 1 presents the general setup of a FMC. It is composed by a set of raw material warehouses $(RMW_1, ..., RMW_n)$ that provide pieces to a set of machines $(M_1, ..., M_n)$. Each machine M_i is fed by the pieces contained in RMW_i, only. The Assembly Units $(AU_1, ..., AU_m)$ gather a certain quantity of machined parts to make an assembled product. The machined parts can be moved from the machines to the assembly units directly or can be stored temporally in the intermediate warehouses $(MI_1, ..., MI_n)$, only if the assembly stations are busy. Every intermediate warehouse is assigned exclusively to every machine. If previous part was stored in an intermediate warehouse, it would have priority over a new machined part. Finally, the products of the assembly units are stored in the departure warehouses $(DW_1, ..., DW_m)$, each one assigned to a assembly station. The FMC includes a set of AGV's $(R_1, ..., R_N)$ that handles all the transportation tasks of the system. Commonly, the intermediate and departure warehouses are located in a unique matrix-shape configuration.

The Hierarchical Control Strategy must be able to assign the transportation task to the AGV, to ensure the convergence of the AGV's to the positions of the warehouses and workstations avoiding the inter-robot collisions. The supervisor ensures that the transportation tasks satisfy the precedence of the machines and assembly units operations, the limitation of the warehouses, the assembly requirements and the product sequences of the FMC.

Next section presents the preliminaries of the FSA formalism, supervisory control theory and the unicycle-type motion control to achieve the control requirements.

III. PRELIMINARIES

III-A. Finite State Automata and Supervisory Control Theory

An event is defined as an asynchronous signal that generates a spontaneous change. DES are systems that depend explicitly on the asynchronous occurrence of events through time. The possible sequences of events in a system are called *language*. Several dynamical systems can be modeled using DES, for instance traffic lights, databases, communication protocols, manufacturing systems,etc. (Cassandras and Lafortune, 2008). The FSA formalism is mathematical structure with a graphical representation of a language. An automaton is defined as a sextuple

$$G = \{X, \Sigma, x_0, X_m, \delta, \Gamma\} \tag{1}$$

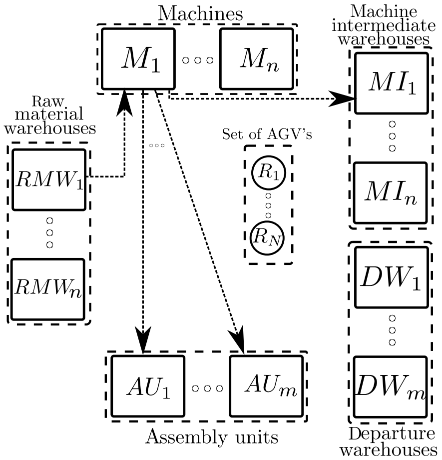
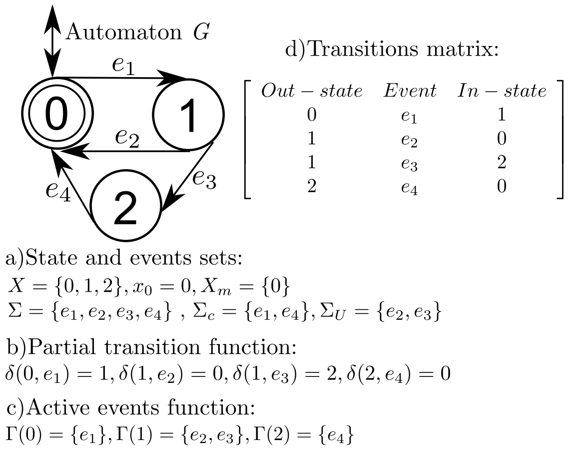


Figure 1. General model of the cell

where X is the set of states , Σ is the set of events, $x_o \in X$ is the initial state,$X_m \subseteq X$ is the subset of marked states,δ : X x $\Sigma \rightarrow X$ corresponds to the partial transition function and finally $\Gamma : X \rightarrow \Sigma$ is the active event function.

Fig. 2 shows an example of *FSA*. The states are represented by circles and the events by arrows. Initial state is the circle with an in-arrow whereas marked states are states with an out-arrow. Note that state 0 is the initial state and a marked state. This automaton is a classical example of the basic operation of a machine, where events e_1 and e_2 indicate the start and stop of a task, respectively, event e_3 represents a machine failure and e_4 indicates that the machine was repaired. Figure 2 also shows the functions δ and Γ. The events can be classified in controllable events (enabled or disabled by and external agent) and uncontrollable events where occurrence can not be forced. In the example of the figure 2, $\Sigma_c = \{e_1, e_4\}$ and $\Sigma_u = \{e_2, e_3\}$ are the controllable and uncontrollable event sets, respectively. Note that the partial transition function also can be translated to the transition matrix (fig. 2 d). This matrix will serve for a mathematical description of the automaton on the control algorithms.

If a plant is composed by the concurrent behavior of several subsystems then, the so-called *synchronous product* allows to mix simple automata of every DES component to synthesize complex models. Let $G_i = \{X_i, \Sigma_i, x_{i0}, X_{im}, \delta_i, \Gamma_i\}, i = 1, 2$ then, the synchronous product of G_1 and G_2 is defined by the new automaton $sync(G_1, G_2) = Ac(X_1 \times X_2, \Sigma_1 \cap \Sigma_2, \delta, \Gamma_{1||2}, (x_{01}, x_{02}), X_{m1}, X_{m2})$ where *Ac* is called the accessible or reachable part of an *automaton G*. In the synchronous product, a common event, i.e. $e \in \Sigma_1 \cap \Sigma_2$, can only be executed if the two automatons both execute it simultaneously. The private events, that is, the events $e \in (\Sigma_1 \backslash \Sigma_2) \cup (\Sigma_2 \backslash \Sigma_1)$ are not subject to such constraint and can be executed whenever possible.



d)Transitions matrix:

$$\begin{bmatrix} Out-state & Event & In-state \\ 0 & e_1 & 1 \\ 1 & e_2 & 0 \\ 1 & e_3 & 2 \\ 2 & e_4 & 0 \end{bmatrix}$$

a)State and events sets:
$X = \{0,1,2\}, x_0 = 0, X_m = \{0\}$
$\Sigma = \{e_1, e_2, e_3, e_4\}$, $\Sigma_c = \{e_1, e_4\}, \Sigma_U = \{e_2, e_3\}$

b)Partial transition function:
$\delta(0,e_1) = 1, \delta(1,e_2) = 0, \delta(1,e_3) = 2, \delta(2,e_4) = 0$

c)Active events function:
$\Gamma(0) = \{e_1\}, \Gamma(1) = \{e_2, e_3\}, \Gamma(2) = \{e_4\}$

Figure 2. Example of a FSA

The behavior of an automaton can present some event sequences that are not desired because they violate a security or process specification or arrive to blocking states. Then, it is necessary to synthesize a *Supervisory Control* (SC) to achieve a given set of specifications. Thus, the plant automaton G represents the uncontrolled behavior of the DES. The plant communicates to the SC the set of feasible events at the state x. Then, the SC enables a subset of events $\Gamma_c(x) \cup \Gamma(x)$ that satisfies the system specifications at this state (fig. 3). For example, if a dispenser in a FMC has capacity of two pieces, the SC must disable the event of arrival of pieces when the dispenser is full and enable it when a slot is available. The supervisor must satisfy the controllability condition, that establishes that only controllable events can be disabled.

The fig. 4 shows the standard procedure to synthesize a SC. The discrete-event behavior of the n components of the plant are modeled by the automata $G_1, ..., G_n$. The plant automata *G* is obtained as the synchronous product of all components. Similarly, the *m* individual specifications of the DES are modeled by the automatons $S_1, ..., S_m$ and their synchronous product results in the total specification automaton *S*. After that, the automaton $G/S = sync(G,S)$, , where *G/S* means the behavior of the plant G controlled by the specification S, performs the supervisor if its language $\bar{K}$ satisfies the controllability condition $\bar{K}\Sigma_U \cap \mathscr{L}(G) \subseteq \bar{K}$, where $\mathscr{L}(G)$ denotes the language of the plant automaton. This condition of controllability, as mentioned above, requires that only controllable events can be disabled and the uncontrollable events must be allowed by the supervisor. If the controllability condition is not satisfied, then it is necessary to remove eventually every state of G/S that violates the controllability condition. Thus, the supervisor is an sub automaton G_c/S that generates the language $K^{\uparrow C}$ which must be controllable. The language $K^{\uparrow C}$ is named as the supremal controllable sub-language. The term supremal indicates that the SC is maximal permissive, i.e. the main objective is to remove the minimum number of states to allow the maximal possible sequences in the system (Wonham, 2009). Note that the last supervisor G_c/S eliminates some events sequences, however it preserves all the original process specifications.

The FSA operations like the synchronous product and the supremal controllable algorithm can be achieved by special DES software as TCT, Supremica, SSPC (Akesson et al., 2006).

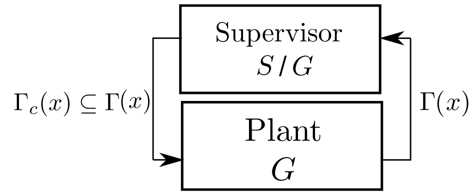


Figure 3. Supervisor Control scheme

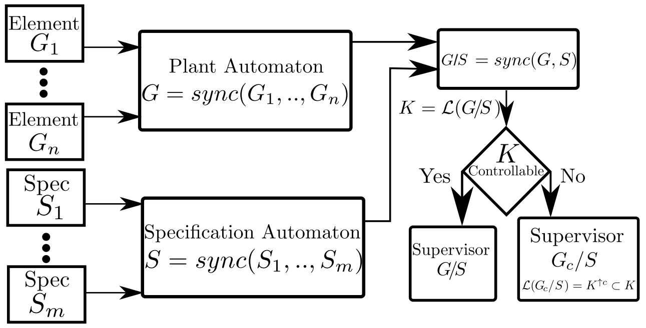


Figure 4. Synthesis procedure of Supervisor

III-B. *Motion control for Unicycle-type robots*

Denote by $N = \{R_1, R_2, ..., R_N\}$ the set of non-holonomic AGVs moving on the cell. The kinematic model for each AGV, shown in fig. 5 is given by

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i & 0 \\ \sin\theta_i & 0 \\ 0 & 1 \end{bmatrix} (\theta_i) \begin{bmatrix} \upsilon_i \\ \omega_i \end{bmatrix}, i = 1, ..., n \tag{2}$$

where υ_i is the linear velocity of the midpoint of the wheels axis and ω_i is its angular velocity. In the rest of the paper the point α_i is considered as the output signal to be controlled given by

$$\alpha_i = \begin{bmatrix} p_i \\ q_i \end{bmatrix} = \begin{bmatrix} x_i + l\cos\theta_i \\ y_i + l\sin\theta_i \end{bmatrix}, i = 1, ..., n \tag{3}$$

The point α_i can be assigned where an actuator is located. This point is chosen instead of the middle point of the unicycle axis since the middle point cannot be stabilized by a continuous time invariant control law (Brockett, 1983). The dynamics of (3) are given by $\dot{\alpha}_i = A_i(\theta_i)\,[\upsilon_i, \omega_i]^T$, where $A_i(\theta_i) = \begin{bmatrix} \cos\theta_i & -l\sin\theta_i \\ \sin\theta_i & l\cos\theta_i \end{bmatrix}$, $i = 1, ..., n$ is the decoupling matrix. The matrix $A_i(\theta_i)$ is non singular, therefore, it is possible to define a control law

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = A_i^{-1}(\theta_i) f_i, i = 1, ..., n, \tag{4}$$

where $f_i = [f_{i1}, f_{i2}]^T$ is an auxiliary control related to the desired dynamics for the point α_i. The control strategy for each AGV for the convergence of the coordinate α_i to a specific desired goal $\alpha_i^* \in \Re^2$, as depicted in the figure 5, is given by

$$f_i = -\frac{1}{2} k \frac{\partial \gamma_i}{\partial \alpha_i} + \sum_{j \in M_i} \varphi_{ij}, i = 1, ..., n \tag{5}$$

where γ_i is an attractive artificial potential function given by $\gamma_i = \|\alpha_i - \alpha_i^*\|^2$, $i = 1, ..., n$ and φ_{ij} is a repulsive vector field defined between the robot R_i and other robot that belong to the subset M_i that contains the position of the AGV's that violates the minimum allowed distance with respect to R_i. Defining $\beta_{ij} = \|z_i - z_j\|^2$, $\forall i, j \in N$, $i \neq j$, then $M_i = \{R_j \in N | \beta_{ij} \leq d^2\}$, $i = 1, ..., n$ where d is the influence zone diameter of each AGV. The set M_i evolves over time due to the movement of AGVs. Thus, the collision avoidance strategy considers the AGVs as mobile obstacles when the robots get close to each other. The function φ_{ij} is commonly designed using the negative gradient of a repulsive potential function. However, in this paper it is selected the repulsive vector field presented in (Hernández-Martínez, 2009) based on a unstable focus given by $\varphi_{ij} = V_{ij} \begin{bmatrix} (p_i - p_j) - (q_i - q_j) \\ (p_i - p_j) + (q_i - q_j) \end{bmatrix}$, where V_{ij} is the repulsive potential function given by

$$V_{ij} = \begin{cases} \eta \left(\frac{d^2}{\beta_{ij}} - 1 \right), & \text{if } \beta_{ij} \leq d^2 \\ 0, & \text{if } \beta_{ij} > d^2 \end{cases} \tag{6}$$

with η a gain parameter that accelerates or slows down the repulsive action. Note that another approach using potential fields can be applied, for instance (Pach et al., 2012).

To ensure finite time convergence and prevent the effects of the death-zone of the actuators, the control law is normalized to $u_i = \frac{\mu}{\|f_i\| + \epsilon} f_i$, $i = 1, .., n$, where μ is the constant of normalization and $\epsilon > 0$ is a small parameter to avoid undetermination when the AGV has converged to its goal.

IV. CONTROL STRATEGY

IV-A. *Hybrid control scheme*

Figure 6 shows the control scheme for the AGV coordination in the FMC. In the high level, a plant model is obtained from the synchronous product of all the elements in the system. Then two supervisors are computed. The plant supervisor enables the process tasks according to some process specifications as logical precedences, sharing resources and limitations of the warehouses. The product supervisor contains the list of events that the plant supervisor must run to manufacture a specific product or a list of products. On the other hand, the transportation tasks are communicated to the AGV's, that apply the control laws given in Section III-B to move the workpieces in the FMC. The next subsection describes the general models to obtain the plant model and the plant supervisor model for a general FMC.

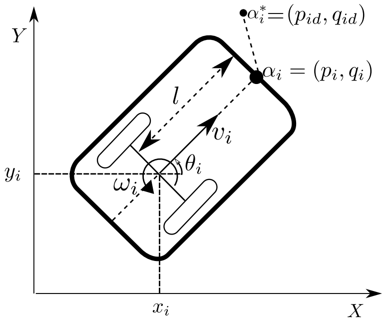


Figure 5. Kinematic model of unicycles

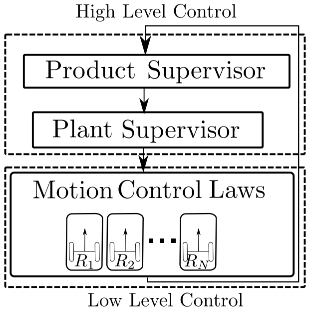


Figure 6. Hybrid Control Architecture

IV-B. *DES Modeling of the AMS*

Applying the ISA-95 standard to all of the process tasks, the assignments can be classified according to the Table I, where start-task and end-task events are denoted. Employing the individual DES models for the AGV's, machines, assembly stations, raw material warehouses and departure warehouses tasks, the plant model is obtained from the synchronization of all the automatons as shown in Fig. 7.

The plant supervisor is obtained according to the specification models displayed in Fig. 8. Until the AGV brings a new raw material piece to a machine, the machine cannot start ($S1$). $S2$ allows the transportation of machined parts to either assembly units or intermediate warehouses until machines have finished. $S3$ models determine the intermediate warehouses capacity. $S4$ denotes that AGV's can

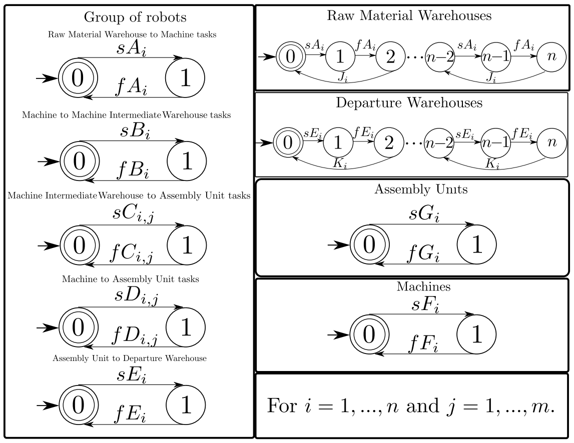


Figure 7. General FSA model for the plant

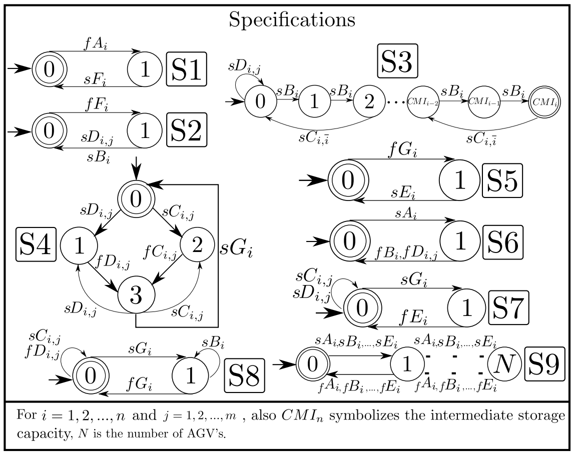


Figure 8. General model of the specifications

place a machined part from either MI_i or M_i to AU_j. $S5$ establishes that AGV's can remove the assembled product once AU_j has finished. $S6$ indicates that AGV's can only bring new raw material pieces when the previous parts of M_i were removed. $S7$ denotes that AU_j can not accept more machined parts from M_i or MI_i until AU_j has not finished tits assembling task. $S8$ establishes priority for the transportation tasks of the machined parts stored earlier over the new parts from the machines to the assembly units. Finally, S9 is a queue model of length n establishing that the n AGV's of the FMC can realize up to n transportation tasks at the same time.

Figure 10 shows an example of a Product supervisor composed only by the start of some tasks that build a complete product sequence in the system. If the plant supervisor is ready to enable some of these process tasks, the Product Supervisor has the priority to select a suitable task in order to achieve desired behaviors.

V. EXPERIMENTAL WORK

Figure 9 presents the case of study, a FMC composed of two Machines, two Assembly Units and two AGV's, i.e. $n = 2$, $m = 2$ and $N = 2$. The tasks of the system are given by the Table II. The plant and supervisor models were computed using the $SSPC$ software (Sánchez et al., 2003).

TABLE I

GENERAL CELL TASKS

Equipment	Task	Start-Event	Finish-Event
AGV	$RMW_i \rightarrow M_i$ $M_i \rightarrow MI_i$ $MI_i \rightarrow AU_j$ $M_i \rightarrow AU_j$ $AU_j \rightarrow DW_j$	sA_i sB_i sC_i sD_i sE_i	fA_i fB_i fC_i fD_i fE_i
Machine	M_i Operation	sF_i	fF_i
Assembly Unit	AU_j Operation	sG_i	fG_i
Raw Material	Arrival to RMW_i	J_i	-
Finished Product	Departure from DW_j	K_i	-
for $i = 1, ..., n$ and $j = 1, ..., m$			

According to the routine shown in the fig. 9, the product supervisor demands two machined parts from M_1 and M_2 to AU_1, AU_2 respectively. Then, the finished assembled product is taken to DW_1 and DW_2, where both products can be withdrawn. The arrows in the fig. 9 represent the transportation tasks demanded to the AGV's by the high level control. When some of these tasks are required, the low level control selects the AGV with the minimum distance respect to the start point of the task and sends it to that location.

The previous routine was tested in an experimental setup composed by two unicycle-type robots model $YRS - A$ manufactured by Yujin Robots, a vision system using a Jai Gigabit Ethernet Camera and the Common Vision Blox software for image processing. Distributed computer systems were implemented, where the vision algorithms stay in one computer connected via RS232 cable to the control computer containing the supervisors models and the continuous control laws using the Real Time Windows Target toolbox of Matlab-Simulink. Figure 10 shows the experimental trajectories of the AGV's in the plane and the figure 11 presents the position of the AGV's recorded by the vision system where the AGV_1 realizes the task A_1 and AGV_2 makes the task A_{12}. Note that the position and orientation of the two AGV's are detected using an ellipse-shape of different dimensions.

TABLE II

CELL TASKS

Task	P.T	Task	P.T
$RMW_1 \rightarrow M_1$	A_1	$M_2 \rightarrow AU_2$	$D_{2,2}$
$RMW_2 \rightarrow M_2$	A_2	$AU_1 \rightarrow DW_1$	E_1
$M_1 \rightarrow MI_1$	B_1	$AU_2 \rightarrow DW_2$	E_2
$M_2 \rightarrow MI_2$	B_2	M_1 working	F_1
$MI_1 \rightarrow AU_1$	$C_{1,1}$	M_2 working	F_2
$MI_1 \rightarrow AU_2$	$C_{1,2}$	AU_1 working	G_1
$MI_2 \rightarrow AU_1$	$C_{2,1}$	AU_2 working	G_2
$MI_2 \rightarrow AU_2$	$C_{2,2}$	New item arrival for RMW_1	J_1
$M_1 \rightarrow AU_1$	$D_{1,1}$	New item arrival for RMW_2	J_2
$M_1 \rightarrow AU_1$	$D_{1,2}$	Product taken out from DW_1	K_1
$M_2 \rightarrow AU_2$	$D_{2,1}$	Product taken out from DW_2	K_2

where $P.T$ means Process Task

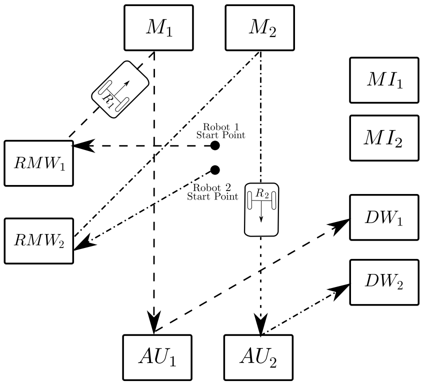


Figure 9. Product Routine

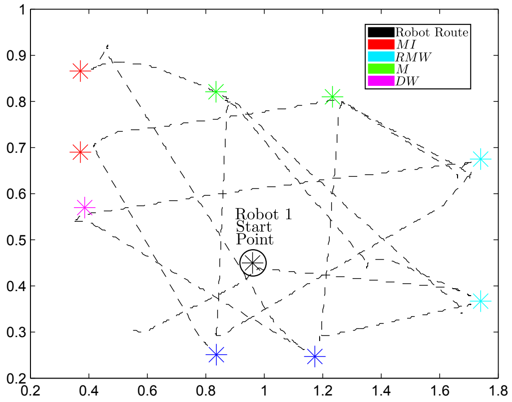


Figure 10. Product Routine Experiment Graph

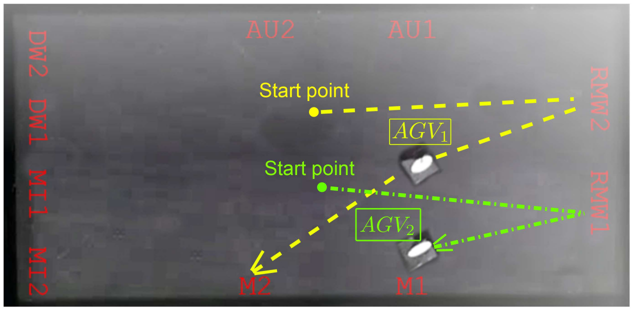


Figure 11. Product Routine Experiment(b)

VI. CONCLUSION REMARKS

A modular hybrid control strategy model has been formulated in this paper for FMC's where a group of AGV's transport the pieces between the workstations and warehouses. The high level provides a methodology to model the DES behavior of the concurrency of the process tasks, transportation tasks and the supervision control according to the process restrictions, logical precedence between tasks and the limitations of the warehouses, separately of the product sequences. In the low-level, the transportation tasks are translated to continuous control laws to ensure the convergence of the AGV's to the desired positions in the FMC. The approach was proved in an experimental platform composed by two AGV's and a vision system to estimate the position and orientation of the AGV's in the workspace. The flexibility of the supervisor when creating several products and the possibility to scale both specification and plant models to any number of workstations in FMC's becomes the approach in a reliable solution for the navigation of AGV's inside manufacturing cells. Other important low-level control functional issues remain to be explored such as emergency procedures, fault recovery and to resolve AGV queueing outside the processing facilities.

REFERENCES

A. Sánchez, E. Aranda-Bricaire, F. Jaimes, E. G. Hernández-Martínez, A. Nava (2009). Synthesis of product-driven coordination controllers for a class of discrete-event manufacturing systems. *Robotics and Computer Integrated Manufacturing*

A. Sánchez, J. Reza, J. Douriet, R. González (2003). A comparison of synthesis tools for supervisory controllers. *Proc. European Control Conference (ECC) Cambridge*.

C. Canudas-de-Wit, B. Siciliano, G. Bastin (2012). *Theory of Robot Control (Communications and Control Engineering)*. Springer.

C. Pach, A. Bekrar, N. Zbib, Y. Sallez (2012). An effective potential field approach to FMS holonic heterarchical control. *ELSEVIER Control Engineering Practice*, vol. 20(12), 1293-1309.

C. Cassandras, S. Lafortune (2008). *Introduction to discrete event systems*. Kluwer Academic.

D. Herrero-Pérez, H. Martínez-Barbera (2010).Modeling Distributed Transportation Systems Composed of Flexible Automated Guided Vehicles in Flexible Manufacturing Systems. *IEEE Transactions on Industrial Informatics*, vol. 6(2), 166-180.

E. G. Hernández-Martínez (2009). *Control Strategies for Multi-agent Systems*. PhD thesis. Mechatronics Section, Cinvestav.

ISA-95 Manufacturing Enterprise Systems Standards CD, Second Edition ANSI/ISA-95.00.03-2005

J. González-Sierra, E. Aranda-Bricaire, E. G. Hernández-Martínez (2011). Strategies with Singularities Avoidance for Groups of Unicycle-type Robots. *subdued to IFAC World Congress*

J. Magallón (2008). *Design and Simulation of an Hybrid Controller for AGV Navigation on Manufacturing Systems*. M. Sc. thesis. Mechatronics Section, Cinvestav.

K. Akesson, M. Fabian, H. Flordal, A. Vahidi (2006).Supremica – A Tool for Verification and Synthesis of Discrete Event Supervisors. *Department of Signals and Systems Chalmers University of Technology* SE-412 96 Goteborg, Sweden

K. Eun-Seok, D. Oron (2012). Multi-location production and delivery with job selection. *In Press, Corrected Proof.*

M. P. Groover (2007). *Automation, Production Systems, and Computer-Integrated Manufacturing, 3rd edition*

P. Farahvash, T. O. Boucher (2004). A multi-agent architecture for control of AGV systems. *13th International Conference on Flexible Automation and Intelligent Manufacturing*, vol. 20(6), 473-483.

R. W. Brockett (1983). *Asymptotic stability and feedback stabilization. In Differential Geometry Control Theory*

S. Makris, G. Michalos, A. Eytan, G. Chryssolouris (2012). Cooperating Robots for Reconfigurable Assembly Operations: Review and Challenges. *45th CIRP Conference on Manufacturing Systems 2012*, vol. 3, 346-351.

W. M. Wonham (2009). *Supervisory Control of Discrete-event Systems, available on website*. University of Toronto,Toronto.

Y. U. Cao, A. S. Fukunaga, A. B. Kahng (1997). Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, vol. 4(1),34-46.

Z. Kovacic, A. Marozin (2003). Simulator of multi-AGV robotic industrial environments. *2003 IEEE International Conference on Industrial Technology*, vol. 2(2), 979-983.