Preprints of the 2013 IFAC Conference on Manufacturing Modelling, Management, and Control, Saint Petersburg State University and Saint Petersburg National Research University of Information Technologies, Mechanics, and Optics, Saint Petersburg, Russia, June 19-21, 2013

Intelligent and Flexible Manufacturing Product Line Supported by Agents and Wireless Sensor and Actuator Network

Edison Pignaton de Freitas*,, Ivan Muller***, Rodrigo Allgayer***,**
Carlos Eduardo Pereira*, Andre Cavalcante****, Marco Marinho**, João Paulo C. Lustosa da Costa**, Rafael Timóteo de Sousa Júnior**.**

** Brazilian Army Technology Center, Brazilian Army, 23020-470, Rio de Janeiro, Brazil. (Tel: +55 21 2410-6274; e-mail: pignaton@ ctex.eb.eb).*
*** Department of Electrical Engineering, University of Brasília, CP 4386, 70910-900, Brasília, Brazil. (Tel: +55 61 3107-5580; {edisonpignaton, jpdacosta, desousa}@unb.br, marco@aluno.unb.br)*
**** Electrical Engineering Department, Federal University of Rio Grande do Sul (UFRGS), CP 15.064 – 91.501-970, Porto Alegre, Brazil (Tel: + 55 51 3316 3561; email: cpereira@ece.ufrgs.br, ivan.muller@ufrgs.br;allgayer.rodrigo@gmail.com)*
*****Electronic and Telcomunication Department, Federal University of Amazonas (UFAM), Manaus, Brazil email: andrecavalcante@ufam.edu.br*

Abstract: The demand for highly decentralized control in factory automation is based on the current needs for flexibility and scalability in the production and assembly in product lines. On one hand, intelligent manufacturing systems, presenting holonic and multi-agent based approaches address these needs, providing the necessary decentralization in the manufacturing control. On the other hand, flexibility and scalability are also related to the disposition of the sensors and actuators devices in the factories, and how easily they can be displaced, introduced and reorganized. This second aspect is addressed by diminishing the needs for cabling to connect these devices, which is possible by using wireless connections instead. This paper explores these two aspects, by studying a flexible product line able to assemble different types of products in a production line by means of a multi-agent approach that drives the products through production units composed of a wireless sensor and actuator network, according to the specification of the products to be assembled. The feasibility of this approach is studied in the performed experiments by assessing timing properties of the proposed technique. This implies in how efficient is the combination of the multi-agent based factory control system with a wireless communication infrastructure in terms of timing.

Keywords: Intelligent Manufacturing System, Multi-agent System, Wireless Sensor and Actuator Network, Holonic Manufacturing System, Distributed Control.

1. INTRODUCTION

Two important current concerns in factory environments are flexibility and scaling, Zampieri (2008). The first refers to the ability to change the product line to assemble new products, or to change the way products are assembled, and the second is related to the overall capacity of the industrial plant. While the first is affected by the logical organization of the factory (e.g. which productions units should be used in the assembly of a given product), the second goes beyond and may require changes in the physical layout, with addition of new production units. Depending on the new products, additional units may be also required to address the first concern.

Distributed control systems are highly desirable in manufacturing systems because they provide flexibility besides other advantages such as increased system robustness, redundancy, and workload balancing, Zampieri (2008). Moreover, distributed solutions are more reliable than centralized systems, which may present a single point of failure and are vulnerable to be overloaded Koptz (1997).

Scaling involves the additional devices to be installed in the factory which may require physical and logical reorganization of the entire product line. These devices must be connected via communication channels either to be controlled by a central control unit (in the case of centralized systems) or to organize themselves in order to perform their tasks (in the case of a decentralized systems). Formerly, these connections were provided by cables which made it difficult to change the factory layout because of the need for production suspensions during long periods and thus, incurring in high costs due to the production outage.

In light of these concerns, different solutions can be proposed. Many intelligent manufacturing systems proposals rely on multi-agent approaches to provide distributed control in product lines, Leitão (2009). There are several benefits in using multi-agents such as the inherent data and control distribution, ability to make heterogeneous devices to cooperate, independent data processing, among others, Wooldridge (2002). To address the problems related to scaling, especially in the addition of new devices and reorganization of the factory layout, wireless connections among the different production units are highly recommended, Liu and Goldsmith (2005). The use of wireless connections avoids the need for cabling to connect devices, making the task of changing their positions inside an industrial environment much easier as well as the insertion on demand of new devices in the network.

Despite the benefits of these approaches to address the specific problems mentioned above, the combination of both in a unified solution may present challenges to make the whole system operational. An important issue in this context is related to the real-time properties of the sensing-control-actuation sequence. In a distributed controlled factory environment driven by multi-agents and interconnected by wireless links, delays in the communication or the migration of the agents among devices may cause system stability.

This paper focuses in a study of how feasible is to meet timing requirements in an intelligent factory system by the combination of multi-agent control systems with wireless infrastructure, particularly exploring the agent migration technique. Migration is a powerful technique in multi agent system (MAS) that allows software agents to start their processes in a given node and finish it in another one, Chess et al (1995). However, migration over wireless connection links may incur in timing disturbances such as delays and jitter, which may negatively affect control systems. This is an important concern that is highlighted in the presented experiments.

The reminding of the paper is organized as follows: Section 2 describes the proposal in combining the multi-agent factory control and the wirelessly connected sensors and actuators infrastructure. Section 3 presents a case study of an industrial plant implementing an intelligent manufacturing production line. Section 4 presents the implementation and experimental results. Section 5 describes related works in the area while Section 6 concludes the paper providing directions for future works.

2. A PROPOSAL TO COMBINE MULTI-AGENTS AND WSAN

The multi-agent support for the industrial scenario considered in this work is inspired on the application of MAS in a canonical wireless sensor network application, targeting tracking, Tseng et al (2003), more specifically in the work presented in Freitas et al (2011). In the tracking target application modelled on that work, once an object is detected in the range of one sensor in the network, a mobile software agent is deployed and it migrates from one sensor to another, following the path traversed by the object. Along the way, the agent draws the path informing it to a control station.

Similarly to the object tracking idea, once a product starts its production, a software agent does also start running in the current processing/production unit, following the product through the units until the product is ready.

Industrial wireless communication systems can be centralized or distributed depending on their characteristics and desired network control. A centralized network is characterized by the existence of a central node responsible for managing tasks and network services, usually called the coordinator. These networks have the disadvantage of being dependent on the coordinator and problems may arise if this node fails. The number of nodes in a centralized network is very dependent of the coordinator processing power because it can be overloaded in a dense wireless network.

In a decentralized network system it is expected that applications and networks have some key features such as self-organization and scalability, in which nodes have to be able to organize themselves independently. To perform self-organization, the developer must provide the capability of automatic addresses and ports distribution for the system, for instance. When it comes to scalability, the network should also be able to communicate with devices that were not previously planned to make part of it, by using a compliant standard interface. Moreover, as the computing effort is distributed, the deployment of additional nodes does not affect the whole system performance because it is not dependent of a central node as it happens in the centralized alternative. Figure 1 depicts a typical distributed wireless industrial communication system in which different kinds of sensor nodes are sensing and collecting physical data. These nodes are able to process this information and transmit them for different nodes responsible for the execution of the control loop. This capability of processing the acquired data makes this type of node more “intelligent” if compared with the sensor nodes typically used in centralized networks, which just sense and send raw data to the coordinator. Due to this capability they are usually called “smart sensors”.

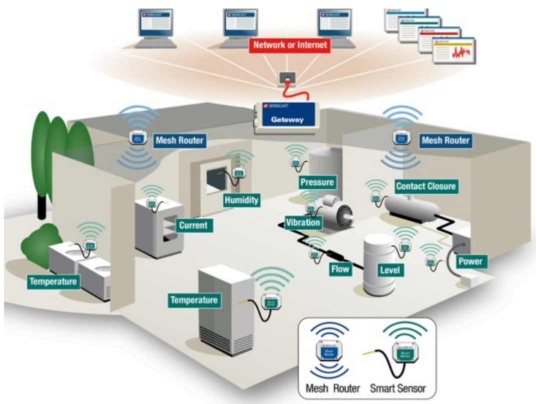


Fig. 1. Distributed wireless communication in an industrial environment, Sensicast (2011).

In an industrial environment in which different products are processed in the same product line, the distributed sensing alternative corroborate to the local interactions of processing/production units in order to provide the necessary support to correctly process each product. However, another important factor is the configuration of the processing/production units themselves, which varies depending on the product to be processed by them. Observing this scenario, a multi-agent approach can help supporting the needed configuration.

Agents carrying the information about how the products have to be handled by each processing/production unit can follow the products during their movement in the production line configuring the machines during the production and assembly process. To perform this task, the agents can exchange information with other agents about parts being produced in other units and those responsible by the production of other products to combine parts or to decide the way a product should take, avoiding jamming, among other possible actions.

The usage of such multi-agent approach can provide the needed decentralized control of the processing/production units (actuators and the sensors) that keep track of the production in the production line and the way they are processed. This alternative solution reduces the communication overhead as there is no need for a central node collecting data from the sensors and sending control messages to all actuator units. It also avoids the need for a preprogrammed control for the whole factory beforehand, which is highly desirable as it provides flexibility to the factory to produce any product at any time without the need to stop the production line to be reprogrammed.

3. A CASE STUDY - INTELLIGENT MANUFACTURING PRODUCTION LINE

The industrial scenario studied in this work is a factory in which the main goal is to assemble different types of products, members of a product line which differ by accessories, and finishing, depending on the specificities of a given product. Starting from a common substrate for the product line, the products are differently handled by different production/processing units according to their specifications. A practical example of such product line can be found in vehicular industry. Car models of a given class may vary depending on number and quality of the accessories that are assembled to a specific model, as well as parts that have to be trimmed differently to comply with specifications. In this type of industry, it is desirable that new products can be introduced into the production without stopping the line. The factory adaptation to provide new outcomes has to be performed so that it continues to be able to deliver the products that were being delivered so far. One way to provide such behaviour to a production line is to attach the specifications close to the product that is been manufactured so that it is recognized by the assembly units while it passes through them. This is done by the actuators performing the actions according to the specifications. Similarly, some products will take different paths through the factory, being processed by the different units according to the specifications. Information about the previous and the next unit that the product passed or will pass is transmitted through the processing units in order to ease next stages, especially when parts have to be merged, follow independent paths inside the plant. For example, this is the case in which there are parts that have to be trimmed in a certain way to fit each other when merged, and in which the production line is supposed to produce different types of products. In this case there is no way to know beforehand how this process should be done. These situations require a distributed control system in which the different production units and the products interact during the production to establish what should be done in each stage.

In this intelligent factory production line, a wireless sensors and actuators network (WSAN) track and interact with the products while they are being produced to determine what has to be done. Once the product starts its way through the production line, a software agent starts running in the first production unit. This agent carries information about the product specification, describing what has to be done in each stage of the production line to deliver that specific product. When the current stage finishes a production process, the agent follows the product to the next stage and so on. During this movement, the software agent migrates from one production unit to another, providing the specification and driving the unit with specific commands to provide the desired result on the final product. After the agent finishes its task in a certain unit, it migrates to the next one to be visited by the product. The great benefit of this approach lies is the fact the entire factory does not need to be previously programmed because it is programmed “on the fly” while the different products are being produced.

When different parts have to be merged, and the processing result of one part affects what is being done in another part, the agent migrates to the unit responsible by the processing of this other part and performs the appropriate actions to guarantee that the two parts will fit each other. This is the case in which an addition of an accessory in a given stage of production requires a specific trimming to fit that accessory in another part in the next production stage, for instance.

As the different units are wirelessly connected, the insertion of a new processing/production unit or the reorganization of the internal factory layout would not compromise the production and could be easily and quickly performed. No cabling would have to be fixed and the production would be resumed in a short time.

Despite the benefits of this approach as mentioned above, some actions in a given production unit may affect other actions that are being performed or those to be performed in other production units. As the whole system is runtime programmed by the agents that migrate from one production unit to another, and as the products move through the units, they are not supposed to wait in a queue, but to be ready for processing when they arrive at a production unit. To make it feasible, timing requirements are extremely important so that there is no risk to create bottlenecks that could compromise the whole chain that could even collapse. Moreover, timing requirements are also extremely important in the interaction between two production units through the wireless link so that a specific task can be performed satisfactorily.

We start with the observation of the specific example in which a part to be assembled in a product has to be trimmed in a specific way to fit the product. In such decentralized controlled factory, an agent controlling a process in a unit may have to migrate (or to send a clone of itself) to another unit to control the trimming of the other part, while still reading and sending commands to the first unit. Thus, two timing requirements have to be considered: 1) the overhead associated to the software agent migration as one production unit should not stop for long periods to wait for commands to process a specific part, and 2) the commands transmitted from one agent to another processing unit has to be timely compliant with the control loop that drives an actuator.

For the second concern, a robotic arm used in the trimming task is considered, which uses a closed control loop, Lages (2008). The robot is able to move and it has a differential drive that can be described by state space model described in Lages (2008) and Allgayer et al (2011).

4. IMPLEMENTATION AND RESULTS

4.1 Implementation and Deployment

For the implementation of the presented case study, the SunSPOT platform, Sun (2010), is used to represent the processing/production units and the mobile platform attached to the product that carries the software agent that controls the production. This makes possible to evaluate the timing overhead to perform agent migration over a wireless link. The agents were implemented using the Agent Factory Micro Edition (AFME) framework, Muldoon et al (2006). AFME is a low scale agent framework, which was developed to enable the creation of planned agents for mobile devices and resource constrained devices. It is designed to handle the Constrained Limited Device Configuration (CLDC)/Mobile Information Device Profile (MIDP) subset of the Java Micro Edition (J2ME) specification. AFME is based on Agent Factory, a large framework for multi-agent systems deployment. The framework is complaint with Foundation for Intelligent Physical Agents (FIPA) specification enabling its interoperability with other FIPA-compliant environments. AFME uses a rule-based concept similar to expert systems, Giarratano and Riley (1989), to represent the agents' behaviors and maintain a reduced set of meta-information about itself and its surrounding environment as the agents' belief. Rules' operations over the belief set determine the agent commitments which finally provide the actions for the agents to perform tasks such as exchange of messages and migration.

The basic AFME infrastructure to run agents in a node is summarily composed of a class implementing the interface `Platform` and an instance of the `BasicRunnable` class from the AFME API. The first provides the basic functionalities to the agents that are hosted in a node, while the second provides the basic functionalities to execute an agent, updating its status and driving the agent's control process according to its rules in order to select the appropriate actions to be performed. In fact, each agent running in a node is represented by an instance of the class `BasicRunnable`. The MAS implementation within mobile agents using AFME requires the basic infrastructure mentioned above and a class implementing the interface `MigrationPlatform`. This interface defines the necessary primitives to move agents from one node to another, as well as to receive mobile agents from other nodes.

The setup for the migration test is composed of two SunSPOT nodes one meter apart from each other, a reasonable distance in the industrial context considered in this work. Both nodes have the previously mentioned AFME platform support and a resident agent running on top of this platform. Besides this software, one of the nodes has also an additional agent which is the mobile agent that will migrate to the other node.

The robot model described by (1) was simulated on the SunSPOT platform and aims to reflect the front of the robot behavior (system output or the result of the actuator trimming a part of a product, for instance) given an input for the next position of the center of mass. The block diagram of the controlled mobile robot system based on (1) is presented in Figure 3. The 100 ms period is considered.

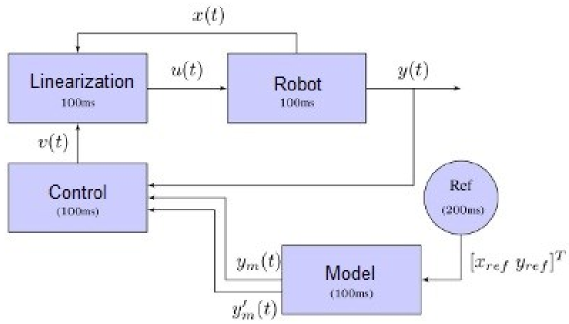


Fig. 2. Block diagram for robot simulation.

The robot system is separated in two distinct nodes: the robot node, executing the actuation on the product, and reference, providing the desired behavior to be compared to the result acquired by the robot. In order to diminish the overhead in the transmission and reception of control messages, a light weight version of RMI (Remote Invocation Method) called mRMI, Allgayer et al (2011), was created to perform this part of the test.

In this simulated system, the reference to the robot is given by:

$$x_{ref}(t) = \frac{5}{\pi}\cos(0.1\pi t)$$

$$y_{ref}(t) = \begin{cases} \frac{5}{\pi}\sin(0.1\pi t) & ,for\ 0 \le t < 20 \\ -\frac{5}{\pi}\sin(0.1\pi t) & ,for\ t \ge 20 \end{cases} \quad (1)$$

The initial point of robot trajectory is $y(t) = (0; 0; 0)^T$.

4.2 Results and Discussion

The agent migration tests evaluated the delay in the agent transmission in which it is possible to separate the associate cost due to the transmission of the agent only, and the agent plus the control messages exchanged by the two communicating nodes to execute the migration. Moreover, in order to better analyse the migration, the cost in terms of number of transmitted bytes is also assessed. Table 1 presents the results (elapsed time to perform the agent migration), while Table 2 presents the number of bytes exchanged during communication between the two nodes.

Table 1. Elapsed Time to Perform an Agent Migration

Measurement considering	Elapsed Time (s)
Agent Transmission Only	1.000357
Agent Transmission plus Control Messages	1.000658

Table 2. Overhead Associated to an Agent Migration

	Overall Size of the Packet Frames (Bytes)			Payload (Bytes)		
Measurement considering	**SN**	**RN**	**Total**	**SN**	**RN**	**Total**
Agent Transmission Only	3378	165	3543	2619	0	2619
Agent Transmission plus Control Messages	3454	235	3689	2639	14	2653
* SN: Sender node; RN: Receiver node						

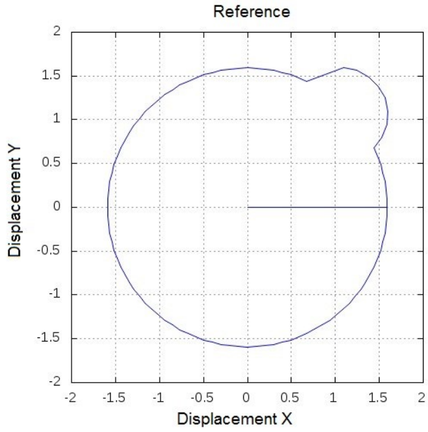
By the observation of the results presented in Table 1, it is clear that there is almost no difference between the time due to the transmission of the agent and the total elapsed time (transmission of the agent plus the control messages), both being close to 1 second. Considering the migration of an agent to configure the machines in a production/processing unit, this time does not present potential to harm the progression of the product in the product line. Additionally, observing the results presented in Table 2, it is possible to state that the number of bytes exchanged between the two nodes due to the control messages does not represent a great overhead compared to the total number of bytes needed to migrate an agent, which can be seen as a positive result.

In relation to the mobile robotic arm control, Figures 3a and 3b respectively present the reference movement and the output provided by the implemented distributed control. Table 3 presents the statistical results for the acquired output considering 200 samples.

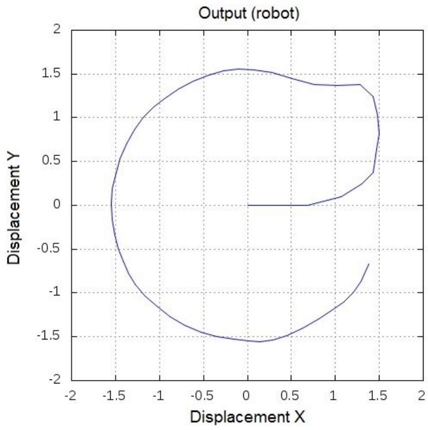
Comparing Figures 3a and 3b one can observe that the distributed control is not able to precisely follow the defined reference trajectory. The results presented in Table 3 are used to explain this control inability. Despite the fact that measurements are close to the goal in average (around 10% more than the target value) the standard deviation is significant, with the value of 27.153 ms. This means that the jitter introduced by the wireless communication between the two nodes is high, resulting in the instability of the output presented in Figure 3b.

Table 3. Statistical results of the robotic arm control

Parameter	**Measurment**
Period	0.100000s
Minimum	0.057000s
Median	0.110000s
Maximum	0.180000s
Mean	0.113744s
Standard Deviation	0.027153s



(a) Reference signal



(b) Output signal

Fig. 3. Distributed control of the robotic arm.

5. RELATED WORK

Agile manufacturing systems were developed on 90's and 2000's years to deal with flexibility, fast deployment and easier configuration of the assembly systems. However, this term comprehends a set of research topics and it serves as a reference for several manufacturing paradigms, such as the Bionic Manufacturing Systems (BMS), by Ueda, 1992, and others; the Holonic Manufacturing Systems (HMS), by Van Brussel et. al., 1998, the Reconfigurable Manufacturing Systems (RMS), by Koren et. al., 1999, and the Evolvable Assembly/Production Systems (EAS/EPS), by Onori, 2002.

The goal of an agile system goes beyond flexibility and rapid configuration as it implies the capability to adapt the system to small and larger changes alike, IDEAS Project, 2010. The goal is to have a fully functional assembly system, without any further programmable effort.

To achieve this, some authors have proposed several directions: Leitão and Restivo, 2006, propose ADACOR, a holonic architecture for manufacturing control that tries to balance between a centralized and a decentralized structure; Ribeiro et. al., 2012, proposes the IADE, a multi-agent environment for flexible production machines that enable the production of several kinds of products simultaneously.

Our proposal has the same direction as IADE, by using a multi-agent system towards a flexible production line. However, besides the concepts of multi-agent systems used in

IADE we also explore the agent migration to provide a fully decentralized and ad-hoc configuration on top of a wireless network infrastructure that links the machines, advancing their proposed approach. Distributed control using wireless links is challenging by itself, as reported in several other works, such as Tanaka et al 2012 and Mohammadi et al 2012. Proposing a distributed control on top of a wireless infrastructure with a mobile multi-agent approach as it is done in our work represents one step further in this challenge. However, as it is shown by the results, it is possible to take advantage of the benefits of the multi-agents to address the control problem in this environment.

6. CONCLUSIONS AND FUTURE WORK

Scalability and Flexibility requirements are highly desired in product line industries which need to adapt and to scale according to the market. Both requirements can be addressed using multi-agent and wireless communication allowing the industry to reduce costs optimizing and customizing the product line manufacturing enabling the production of a largest number of different products.

In this paper timing properties are assessed in order to evaluate the suitability of the combination of the multi-agent approach with mobile agents running on top of a wireless network infrastructure. Two different experiments were performed, the first one to evaluate the agent migration itself and the second one to evaluate the distributed control over a wireless link. The results indicate the applicability of the proposed ideas, despite the degradation in the performance of the distributed control.

A clear direction for future works in this area is the proposal of new wireless communication protocols to enable jitter reduction, a sensitive variable to critical control systems. In this context, TDMA based data link layer algorithms must be considered, as they are able to cope with jitter. Other advanced multi-agents techniques can also be explored, such as agent cloning. Models exploring this technique can also be mentioned as possible directions for future works.

REFERENCES

Allgayer, R., Muller, I., Pereira, C., Cavalcante, A., Pignaton, E., Larsson, T. (2011). Analysis of distributed control system using remote method invocation in wireless network, Proceedings of the 37th Annual Conference on IEEE Industrial Electronics Society, 346-351.

Chess, D. Grosof, B. and Harrison, C. (1995). *Itinerant agents for mobile computing*. Technical Report RC 20010, IBM Research.

IDEAS Project (2010). FP7 - Instantly Deployable Evolvable Assembly Systems (IDEAS) Project. Royal Institute of Technology.

Freitas, E. P., Bösch, B., Allgayer, R. S., Steinfeld, L., Wagner, F. R., Carro, L., Pereira, C. E. and Larsson, T. (2011). Mobile Agents Model and Performance Analysis of a Wireless Sensor Network Target Tracking Application. In: S. Balandin, Y. Koucheryavy, and H. Hu. (Org.). *Proceedings of 10th IEEE International Conference on Next Generation Wired/Wireless Advanced Networking (New2An'11)*, 274-286.

Giarratano, J.C. and Riley, G. (1989). *Expert Systems: Principles and Programming*. Brooks/Cole Publishing Co., Pacific Grove, CA.

Koptz, H. (1997) *Real-time systems: design principles for distributed embedded applications*. The Springer International Series in Engineering and Computer Science, Springer.

Koren, Y. et. al. (1999) Reconfigurable Manufacturing Systems. CIRP Annals – Manufac. Tech., 48, 527-540.

Lages, W.F. (2008). *Real-Time Project: Mobile Robot Control*. Technical report, UFRGS, Brazil.

Leitão, P. (2009). Agent-based distributed manufacturing control: A state-of-the-art survey. Engineering Applications of Artificial Intelligence, 22, 979-991.

Leitão, P., Restivo, F. (2006) ADACOR: A holonic architecture for agile and adaptive manufacturing control. Computers in Industry, 57, 121-130.

Liu, X. and Goldsmith, A. (2005). Wireless network design for distributed control. *Proceedings of the 43rd IEEE Conference Decision and Control*, vol. 3. 2823–2829.

Mohammadi, A.; Tavakoli, M.; Marquez, H.J. (2012) Control of nonlinear teleoperation systems subject to disturbances and variable time delays. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 3017-3022.

Muldoon, C., O'Hare, G.M.P., Collier, R., O'Grady, M. J. (2006). Agent Factory Micro Edition: A Framework for Ambient Apps. *Comp. Science*, Springer, 727-734.

Onori, M. (2002) Evolvable Assembly Systems – a new paradigm? 33rd International Symposium on Robotics (ISR). Stockholm: Intl Federation of Robotics, 617-621.

Ribeiro, L. and Rosa, R. and Cavalcante, A. and Barata, J. (2012) IADE – Ideas Agent Development Environment: Lessons Learned and Research Directions Proceedings of the 4th CIRP Conference On Assembly Technologies And Systems (CATS), Michigan: CIRP.

Sensicast (2011) Remote monitoring solutions Website, accessed in: October, 2011. [Online]. Available: http://www.sensicast.com

Sun Microsystems, Sun Small Programmable Object Technology - Theory of Operation. (2010) Available at: <http://www.sunspotworld.com/docs/index.html>. Accessed in: October, 2010.

Tanaka, K., Tanaka, M., Ohtake, H., Wang, H.O. (2012) Shared Nonlinear Control in Wireless-Based Remote Stabilization: A Theoretical Approach, IEEE/ASME Transactions on Mechatronics, , 17, 3, 443,453.

Tseng, Y. C., Kuo, S. P., Lee, H. W. and Huang, C. F. (2003). *Location tracking in a wireless sensor network by mobile agents and its data fusion strategies. Information Processing in Sensor Networks Book*, Springer.

Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P. (1998) Reference architecture for holonic manufacturing systems: PROSA . Computer in Industry, Elsevier, 37, 255.274.

Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*, John Wiley & Sons Ltd.

Zampieri, S. (2008) Trends in networked control systems. *Proceedings of 17th IFAC World Congress*, 2886-2894.