Hybrid Control Models for Flexible Manufacturing Systems

L. Mic, R. Fat, M.M. Santa, T.S. Letia
Department of Automation, Technical University of Cluj-Napoca, (UTCN)
Cluj-Napoca, Romania
Liana.C.Mic@gmail.com, Raluca.Fat@staff.utcluj.ro, Maria.Santa@aut.utcluj.ro, Tiberiu.Letia@aut.utcluj.ro

***Abstract*—Flexible Manufacturing Systems (FMSs) need Hybrid Control Systems (HCSs) to solve the specified requirements. An HCS is composed of a discrete event system (DES) and a continuous Discrete Time System (DTS) that interact each other to fulfil some shared goals. The control of hybrid systems is difficult due to the complex requirements concerning asynchronous reactions to discrete events and continuous adjustment of some outputs. The main goals of the current research are to conceive models that describe the structure and the behaviour of an FMS (which by its nature is a hybrid system) and the synthesis of HCS that determines the desired behaviour with specified performances. The proposed models endow the Enhanced Time Petri Nets (ETPN) with Fuzzy Logic (FL) inference rules obtaining the Fuzzy Logic Enhanced Time Petri Net (FLETPN) models. These can be used to describe the control and controlled components of an FMS. Components that include FLETPN models are used to partition the HCS with the aim to divide the complex problem into smaller sub-problems that can be solved step by step using genetic algorithms.**

Keywords—Flexible Manufacturing Systems; Hibrid Control Systems; Fuzzy Logic Enhanced Time Petri Nets; Genetic Algorithms.

I. INTRODUCTION

The development of FMSs by the introduction of robotic structures represent new directions for research and improvement of the performances of production lines. The main activities that can be performed by the robots included in FMSs are bounded to objects manipulation of components or sub-assemblies. New tasks impose production planning using cooperation between the machines performing assembling processes and robots.

An FMS is a production system capable to adapt to different production tasks from the shape and dimension features to the technologic process that have to be engaged.

The main components of FMSs are:

- *Flexible manufacturing unit*: representing a complex machine, multifunctional equipped, an automated handler that can function automatically.
- *Flexible manufacturing cell* composed of two or more flexible manufacturing units with machines controlled directly by computers.
- *Flexible manufacturing system* consisting of multiple manufacturing cells (group of machines with numerical command) connected through automated systems and all of them being under the direct control of a computer that orchestrates and coordinates the sub-systems by means of a computer network.

The system training for making decisions is a difficult task due to the FMS complexity and the distribution of the control system in different components. The modification of requirements leads to the reconfiguration of the entire system.

The objective of the current research is to elaborate models that describe the functions performed by the FMS, such as the canting operation or the drilling operation or the human interaction with the system to manually achieve some operations affecting the control system or the controlled machines. The description of the FMS at the system level uses UML component diagrams. They partition the entire system into interacting subsystems. The high level components can include lower level components that integrate a new kind of Petri nets that describe the operations performed by machines and robots components, incorporating the applicable criteria that follow.

II. RELATED WORKS

The approach proposed in [1] tackles the problem of multi-periodic design of the Dynamic Cellular Manufacturing Systems and proposes a new mathematical model. The most important problems being addressed by the proposed model are: machine duplication, capacity of the machines, operations sequencing, alternative processing ways of the products, varying requirements of products, lots division etc. It makes decisions of most of the problems that arise in the system, as the formation of the cells, intra and inter cellular layout, products routing and products flow between machines. The solutions are presented as two approaches, one combines Simulated Annealing with Linear Programming and the other Genetic Algorithms with Linear Programming.

In [2] is proposed a multi-objective model for a cellular manufacturing system that operates in a dynamic environment. The proposed algorithm takes into calculus the design problems of the manufacturing cells such as the assignment of products to the machines and load balancing based on the

978-1-5386-2205-6/18/$31.00 ©2018 IEEE

operational time and operations sequencing for the creation of the parts. A multi-objective optimization algorithm is proposed for elements extraction and tracking constraints handling.

In [3] is proposed a hybrid technique for errors handling based on two knowledge models: fuzzy generalized rules of the form event-condition-action and fuzzy typed Petri Nets extended with process knowledge. Realizes integrated representations and reasoning with fuzzy and non-fuzzy knowledge as well by the knowledge of the application domain and of the process workflows. Besides these it also supports two strategies of manipulation during the exceptions handling: direct and analysis-based decision making. Lastly is developed a fuzzy reasoning algorithm that uses probabilities to address the problems of reasoning of the uncertain objectives and of the concepts of certain goals.

Reference [4] presents an approach for the modelling and analysis of complex, dynamic and time-critical systems using stochastic Petri Nets together with fuzzy sets. The proposed method is organized on two levels: one is for the stochastic Petri Nets with one difference that the states probabilities are obtained parametrically under the form of transitions executions rates, and on the second level these rates are described using triangular fuzzy numbers and the probabilities of the steady states are finally calculated. The importance of the proposed method is that it takes into consideration both dimensions of the uncertainty, stochastic variability and imprecision.

In [5] are considered the problems of machine allocations and determination of the facilities, which are a branch of the general problem of facilities determination in which is done the selection of the machines location and is determined the route of processing for each product. In this study the requested production volume was expressed using fuzzy numbers with different functions. In order to solve the problem the deterministic model is integrated with a fuzzy implication by means of the expected value model. Finally an intelligent hybrid algorithm was developed based on a genetic algorithm and a fuzzy simulation method and it is applied.

Reference [6] propose an intelligent framework for the design of robotic flexible assembly systems. The method is based on the knowledge formalisms with Petri Nets that use them together with knowledge-based systems techniques. The assembly system is modelled and analysed using a formal representation based on knowledge Petri Nets. The results show that the proposed approach can be applied for the design, simulation, analysis, evaluation and optimization of the system's layout inside an intelligent integrated environment.

In [7] is made a general presentation of some of the principles from the nature and biology and is analysed the efficacy of these methods used in the development of multi-agent systems to solve difficult problems in the manufacturing field. The most important between these technologies are: Genetic Algorithms, Particle Set Optimization, Ant Colony Optimization, and Artificial Bee Colony. In order to prove the applicability of the bio-inspired techniques in the domain of manufacturing an experiment was done with a FMS where the control system used in its functionality bio-methods and had as objectives the dynamic allocation of tasks and the dynamic routing of pallets.

Reference [9] presents a synthesis of the current state-of-the-art from the domain of control of the manufacturing systems which use techniques from the Artificial Intelligence, as multi-agent systems or holonic manufacturing systems. The present requirements from the industry and economy are to construct manufacturing systems that respond efficiently and effectively to the dynamic changes inside the environment and at the same time keeping their productivity and quality. Thus there is a big need for intelligence, adaptability, reliability. The paper also discusses the reasons for the weak adoption of those techniques in the industry at present.

The same problem is approached in [8]. Here it is proposed a multi-agent system used in production planning from the reconfigurable enterprises constituted of complex production capacities, articulated and geographically dispersed. These are reconfigurable production systems characterized by rapid adaptability of the production capacity and of functionalities to construct different types of products.

In [10] a methodology for the scalability problem that can be applied in the reconfigurable manufacturing systems and that can incrementally scale the production capacity of the system is presented. It is developed an algorithm based on the technique of Genetic Algorithms to determine the most economical path to reconfigure the system such as the addition or removal of machines in order to fulfil new requirements and the re-balancing of the system for each new configuration. The proposed technique is validated into a system of propulsion of cylinders into a car.

III. Fuzzy Logic Enhanced Time Petri Nets

According to [11] a FLETPN is defined as:

FLETPN = (P, T, pre, post, Inp, Out, D, W, X, EFS, FLRS, α, β, M),

Where: $X = \{x_1, x_2, ..., x_m\}$; $x_i \in [-1, 1]$ □ □**,** with □ the real number set,

W: P × T → **R;** $W(p_i, t_j) \in$ □**,** waiting coefficients,

α: P → X, a mapping that assigns the variables to places,

β: T → **FLRS,** a mapping that assigns fuzzy logic rule sets to transitions.

Inp □ P, Out □ P sets of input and output channels linked to Petri net places.

The Fuzzy Set: FS = {NL, NM, ZR, PM, PL} is extended with Φ the empty set meaning "it is not known" at the current moment of time. The result is the EFS (extended fuzzy set).

FLRS = $\{FLRS_1, FLRS_2, ..., FLRS_k\}$, FLRS is the set of all fuzzy logic rule set, where $FLRS_i$ is a set of fuzzy logic rules assigned to a transition t_i.

The FLRSs of different transitions are not compulsory to be distinct. Tab. I contains an example of FLRS. The blank character states the corresponding rule is missing.

TABLE I. FLRS EXAMPLE

x1/x2	NL	NM	ZR	PM	PL
NL	NL,PL	PM,NL	ZR,ZR	NL,PL	ZR,PL
NM	PL,PM	NM,ZR	PL,NM	PL,NM	NM,PL
ZR	NL,PM	PL,NM	ZR,ZR	R,NM	PL,ZR
PM	ZR,PL	ZR,PM	NM,PM	PM,PM	NL,NL
PL	PM,ZR	PM,NM	ZR,ZR	NM,ZR	PL,NM

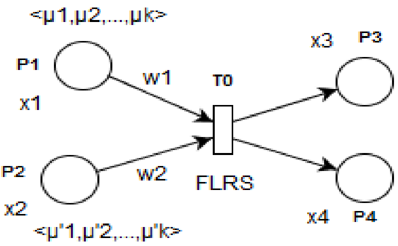

Fig. 1. FLETPN example

The token set S is:

$S = \{<\mu_{-2}, \mu_{-1}, \mu_0, \mu_1, \mu_2> \mid \text{for all } k, \mu_k \in [0,1], \sum_k \mu_k = 1\}$,

Where μ_k (k=1, 2, ..., 5) are the membership degrees of the variable x_i (assigned to a place p_i) to a fuzzy logic value from the set FS.

The marking is: $M : P \rightarrow S \cup \Phi$,

$M(p_i) = <\mu_{-2}, \mu_{-1}, \mu_0, \mu_1, \mu_2>$ or Φ. The marking $M(p_i) = 0$ from the classical Petri net corresponds here to $M(p_i)= \Phi$.

A place can contain no token (i.e. Φ*)* or only one token. If a place contains a token and another one has to be set at the current moment of time, the last token will replace the previous one.

A transition t_i is enabled if and only if there is at least one rule in its $FLRS_i$ that can be activated with the content of its input place set at the current moment of time.

If more than one rule can be activated at the current moment of time, all the rules are used and the resulted tokens is the sum of the tokens. A normalization procedure is applied so that an output place contains one token that met the requirement:

$\mu_{-2} + \mu_{-1} + \mu_0 + \mu_1 + \mu_2 = 1$.

The execution of an enabled transition t_j involves:

- The extracting of the tokens from the transition input places (denoted by $^o t_j$).
- The defuzzification of all input variables x_i.
- The multiplication of the variables with the corresponding weighting coefficients $x_{ij} = w_{ij} \cdot x_i$;
- the fuzzification of the variables x_{ij} ;
- The use of the FLRS with x_{ij} as inputs;
- The normalization operation that reduces the previous consequences to a single one and leads to the setting of a single token into the output places;
- The setting of the resulted tokens into the transition output places (denoted by $t^o{}_j$) when the delay elapses;
- If a transition is started, another (of its) start can be engaged only after the previous execution ends.

IV. PROBLEM DESCRIPTION

The main objectives of the current research are the formal specification and modelling of the components, operations and functionalities of an FMS. FMSs are complex systems conceived of multiple machines that can execute hundreds of types of operations with complex iterations. An FMS can contain between 5 and 20 machines and can implement over a thousand operations.

One of the most important features is their capacity of adaption to new changes in production type, requirements, functionalities etc. The FMS specifications concerns its structure (describing the machines, robots, conveyors), the functions (operations) of the components, operations parameters, processing accuracy etc. A component diagram of an FMS is described in Fig.2.

As can be seen in the Fig.2 every component in the system is controlled by the Control Manager (CM). Each machine M_i, (i=1, …, n) and robot R_j, (j=1,…, m) have their proper controllers C_k, (k=1, …, n+m) which interacts with the CM through interfaces that include input and output methods. CM demands the local controller machine to perform different activities using methods of the Outset and the local controller uses methods of the CM Inset to signal the end of the requested operations. CM receives a technology from the user and schedule it for execution on a chain composed of machines and robots.

The objectives of the current research concern the communication between CM and local controllers as well as the interactions between local controllers and machines or robots.

An example of an FMS is a system consisted of five machines and two robots. The robots have as main tasks to move the parts of material from one machine to another in order to be processed. Each machine has a certain functionality: Severing, Canting, Milling, Turning, and Drilling, as can be observed in Table II.

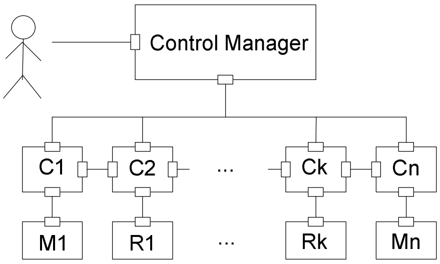

Fig. 2. Component diagram of a FMS

TABLE II. MACHINES AND THEIR OPERATION AND ROBOTS

Operation	Machine	Robot
Severing	M1	R1
Canting	M2	R1
Milling	M3	R1,R2
Turning	M4	R2
Drilling	M5	R2

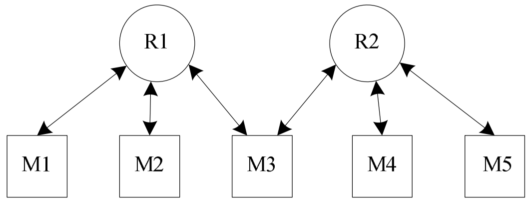

Fig.3. Structure of an FMS example

For this example, the FMS structure is presented in Fig. 3. The robot R1 moves the parts of material in order to be processed by the machines M1, M2 and M3, and the robot R2 moves the parts to be processed by the machines M3, M4 and M5. The robots and the machines have local controllers that receive commands from the CM.

A. *Technology descriptions*

For the achievement of a product p_i, the product technology are defined by the activities scheduling. An activity A_j is described by $A_j(par_j)$, where par_j are the parameters of activities. The activities are denoted by the operators: * (sequence), & (concurrency), + (selection), # (loop). An example of a product technology is described by the expression:

$$\tau(p_i) = A_1(par_1) * (A_2(par_2) + A_3(par_3) \& A_4(par_4)) * A_5(par_5) \dots A_j(par_j) \quad (1).$$

B. *Activities and subactivities*

An activity A_j can have more sub-activities a_i such as one determined by three parameters x, y, z:

$$A_j(par_j) = a_1(x_1,y_1,z_1) * a_2(x_2,y_2,z_2) + a_3(x_3,y_3,z_3) * a_4(x_4,y_4,z_4) \dots a_k(x_k,y_k,z_k) \quad (2).$$

An activity is performed by an assigned component. When a component receives a request to perform an activity A_j, the parameters are read and the corresponding task is started.

The task implements a sequence as (2) that includes sub-activities. An activity can be considered the drilling activity. For the drilling activity A_j the sub-activities a_k are:

- catch (),
- (move(x_n) & move(y_n)) * startDrill() * move(z_n) * haltDrill() * move(z=0) * move(x=0) * move(y=0);
- release().

V. DEVELOPMENT OF CONTROL SYSTEMS

The component diagram presented in Fig. 5 is detailed further for each activity. For example the drilling activities. Relay ports are represented by a cycle in a square and contains one location and one transition with arrow sense. The Drilling Machine (DM) receives the command from D-controller. The command are: catch, move x, move y, move z, drill and release.

In Fig.4 is represented the FLETPN for the M_x component. Similarly can be described the M_y and M_z component. For M_x component x_r is the requested x and x_c is the current x.

Notations:

- $move_x(-1)$ – move to $x < x_c$;
- $move_x(1)$ – move to $x > x_c$;
- $move_x(0)$ – stop;
- drill(1) – start drill & move z;
- drill(0) – stop(exit the drill).

For M_x component the FLETPN expression is:

$$\tau_x = t1(x_c, mvx) * t2(x_c, mvx)* \quad (3).$$

DM inserts continuous the x_c positions. Notation mvx means move(x) and

- if $x_r < x_c$ then mvx = -1;
- if $x_r > x_c$ then mvx = 1;
- if $x_r = x_c$ then mvx = 0.

Transition t1 contains the differences rules set between x_r and x_c, as in Table III and similarly transition t2 contains the comparator rules set between x_r and x_c as in Table IV.

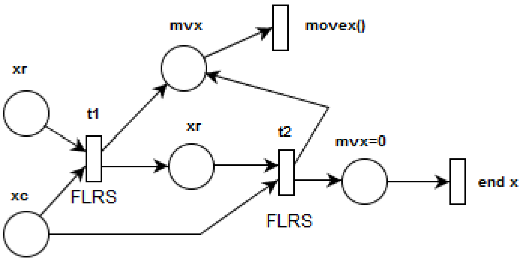

Fig. 4. The FLETPN for the M_x Component

TABLE III. THE DIFFERENCES TABLE FLRS

x1 \ x2	***NL***	***NM***	***ZR***	***PM***	***PL***
NL	ZR	NM	NL	NL	NL
NM	PM	ZR	NM	NL	NL
ZR	PL	PM	ZR	NM	NL
PM	PL	PL	PM	ZR	NM
PL	PL	PL	PL	PM	ZR

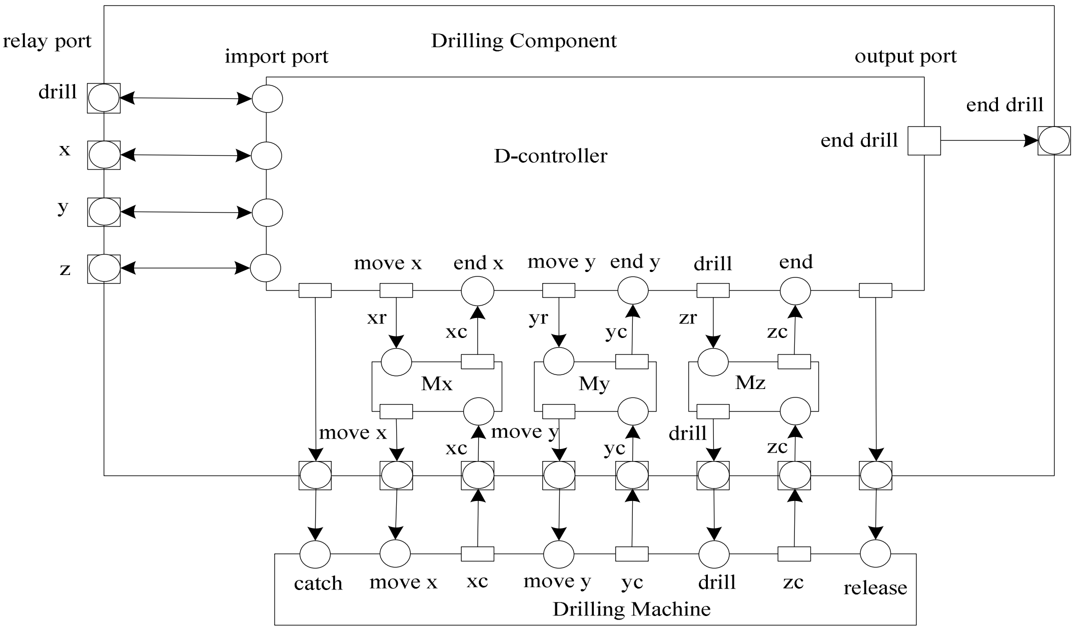

Fig. 5. Drilling Component Diagram

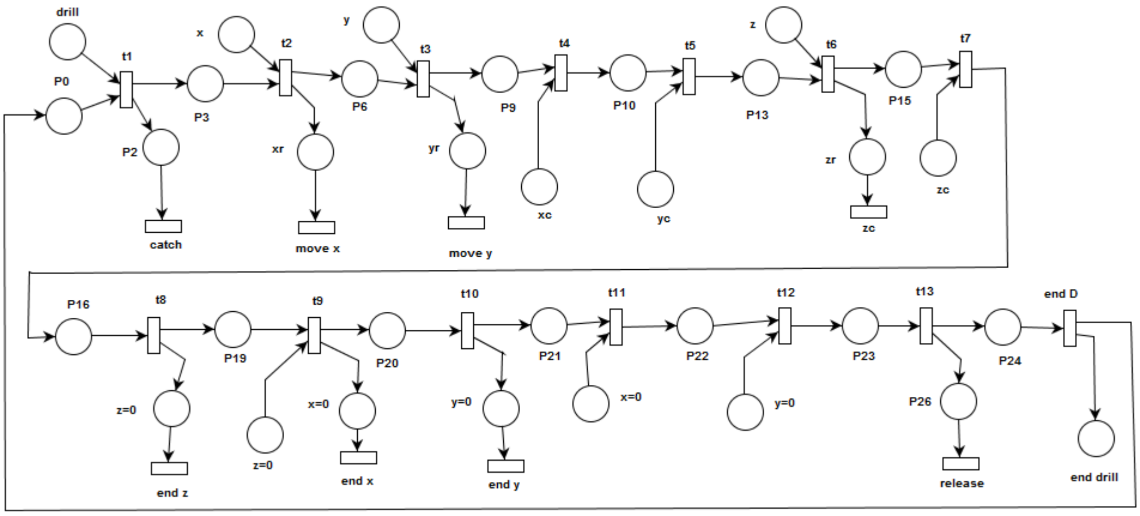

Fig. 6. The FLETPN for the D-Controller Component.

The FLETPN for the D-Controller Component is represented in Fig. 6. The transitions contains FLRS table.

For D-Controller the expression is:

$\tau d = catch() * (move_x(x_r) \ \& \ move_y(y_r)) * drill(z) * drill(0) * (move_x(0) \ \& \ move_y(0)) * release()$ (4).

And the FLETPN expression:

$\tau d = t_1(\phi_1, catch) * t_2(x, x_r) * t_3(y, y_r) * t_4(x_c, \phi) * t_5(y_c, \phi) * t_6(z, z_r) * t_7(z_r, \phi) * t_8(\phi, 0) * t_9(x_0, 0) *$ (5).

The InpSet = $\{x, y, z, x_c, y_c, z_c\}$ and

the OutSet $= \{$catch, release, $x_r, y_r, z_r\}$.

TABLE IV. THE COMPARATOR TABLE FLRS

x1 \ x2	***NL***	***NM***	***ZR***	***PM***	***PL***
NL	NL	NM	ZR	PM	PL
NM	NM	NM	ZR	PM	PL
ZR	ZR	ZR	ZR	PM	PL
PM	PM	PM	PM	PM	PL
PL	PL	PL	PL	PL	PL

VI. CONCLUSION

Advantages of the proposed models:

- The models contain only one type of token even if the places contain different kinds of variables.

- The mappings of the transitions link different kinds of variables.
- The mappings play two roles. One to enable the execution of the transitions, and another to determine the variables that are set in the transition output sets.
- There is no need to add extra conditions for the execution of the transitions. They are included in the tables describing the mappings.
- The ranges of the variables are clearly specified and can be verified.
- The temporal behaviour is modelled and it can be determined by the internal or external conditions.
- The description is flexible and can be extended to more complex relations between variables.
- The models describe the structure and the behaviour too. That means: the verification, the simulation and the implementation are achieved using only one kind of model.
- The development based on components that include FLETPN models allow to tackle complex and large systems due to the possibility to construct components that aggregate others components.

The drawbacks of the proposed models are:

- The model synthesis is more complicated compared to the classical Petri nets. The current research used an interactive development. One activity concerns the structure generation and another one focusses on the endowing the transitions with the FLRSs.
- The verification of the FLETPN models has to be split on two phases. The first one analyses the structures as in the classical Petri nets and the second phase involves the effects of the FLRSs on the dynamical behaviour. [

REFERENCES

[1] H. Bayram, R. Sahin, “A comprehensive mathematical model for dynamic cellular manufacturing system design and Linear Programming embedded hybrid solution techniques”, Elsevier, Computers&Industrial Engineering, 2015, pp. 10-29.

[2] H. Nouri, “Development of a comprehensive model and BFO algorithm for a dynamic cellular manufacturing system”, Elsevier, Applied Mathematical Modelling, 2015, pp. 1-18.

[3] Y. Ye, Z. Jiang, X. Diao, G. Du, “Extended event-condition-action rules and fuzzy Petri Nets based exception handling for workflow management”, Elsevier, Expert Systems with Application, 2011, pp. 10847-10861.

[4] F. Tuysuz, C. Kahraman,”Modeling a flexible manufacturing cell using stochastic Petri Nets with fuzzy parameters”, Elsevier, Expert Systems with Applications, 2010, pp. 3910-3920.

[5] H. Nasab, “A hybrid fuzzy GA algorithm for the integrated machine allocation problem with fuzzy demands”, Elsevier, Applied Soft Computing, 2014, pp. 417-431.

[6] X. Zha, H. Du, Y. Lim, “Knowledge intensive Petri Net framework for concurrent intelligent design of automatic assembly systems”, Pergamon, Robotics and Computer Integrated Manufacturing, 2001, pp. 379-398.

[7] P. Leitao, J. Barbosa, D.Trentesaux, “Bio-inspired multi-agent systems for reconfigurable manufacturing systems”, Elsevier, Engineering Applications of Artificial Intelligence, 2012, pp. 934-944.

[8] M. Bruccoleri, G. Lo Nigro, G. Perrone, P. Renna, S. Noto la Diega, “Production planning in reconfigurable enterprises and reconfigurable production systems”, Universita di Palermo, 2012, Italia.

[9] P. Leitao, “Agent-based distributed manufacturing control: A state-of-the-art survey”, Elsevier, Engineering Applications of Artificial Intelligence, 2009, pp. 979-991.

[10] W. Wang, Y. Koren, “Scalability planning for reconfigurable manufacturing systems”, Elsevier, Journal of Manufacturing Systems, 2012, pp. 83-91.

[11] T. S. Letia and A. O. Kilyen. “Fuzzy Logic Enhanced Time Petri Net Models for Hybrid Control Systems”, Proc. of AQTR 2016 IEEE Conference, Cluj-Napoca, Romania, pp. 1-6, doi: 10.1109/AQTR.2016.7501322.