Machine Learning Based Manufacturing Control System for Intelligent Cyber-Physical Systems

Vaibhav Shah
Assistant Professor
University of Minho
School of Engineering
Department of Information Systems
Portugal

Goran D. Putnik
Full Professor
University of Minho
School of Engineering
Department of Production and Systems
Portugal

Cyber-physical systems are often misunderstood to be just any embedded systems. The real cyber-physical system should have both physical and digital (computational-communication-control) parts inter-connected in each part and process, and the system itself should have the capacity to change its own behaviour to adapt to changing requirements. This paper presents an architecture of an intelligent cyber-physical system where a reconfigurable manufacturing system is supported by a machine learning algorithm to provide enhanced decision-making to the manufacturing control system. Experiment results are presented showing the machine learning module can help the control system adjust itself with changing requirements provided externally (by a user) in the form of training examples. The result is an architecture of an intelligent cyber-physical system, with physical and digital parts always working in synchronization, enabling change in the system's behaviour in terms of manufacturing process-flow in order to adapt to any change in the production planning.

Keywords: *Industry 4.0, machine learning, reconfigurable manufacturing systems, cyber-physical systems, intelligent systems, control systems, grammatical inference.*

1. INTRODUCTION

Traditional manufacturing systems have evolved a long way in the past few decades, from static standalone lathe machines, to highly reconfigurable interconnected ubiquitous manufacturing systems [1]. With the technological revolution envisaged in the Industry 4.0 [2], and high demand both in terms of the quantity and personalised products, reconfigurable manufacturing systems have become almost a fundamental requirement for sustainable growth. There are many approaches for reconfigurable manufacturing systems, as described in a recent systematic review [3]. However, as a basic requirement, the reconfigurable manufacturing systems must have adjustable structure on the hardware and software levels [4], [5]. This implies there needs to be a control system to enforce and 'validate' these adjustments or reconfigurations. The manufacturing systems, irrespective of their type or domain, do have a control system. However, in order to have a more dynamic and highly reconfigurable manufacturing system, an enhanced control system with intelligent decision making is needed.

On the other hand, cyber-physical systems (CPSs) are essential manifestations of Industry 4.0 and, at the same time, one of its essential instruments as well. Another essential component of advanced CPS is artificial intelligence (AI) which, when embedded in CPS, enables intelligent CPS, i.e. I-CPS. A cyber-physical system, as defined by the author who first coined the term, is not just an embedded system with hardware and software systems communicating. A true cyber-physical system must have a cyber (computing, communication and control) component included in every physical component [6]. The inherent characteristic of the cyber-physical systems is such that they do not preserve the determinism in the logical and control structure of the system [7]. Hence, the system should have in-built capacity to allow and manage the change in the operational structure dynamically.

In order to include both the above features in a single system, i.e. to create a manufacturing system with the ability to reconfiguration with intelligent decision making, as well as integration of computing and communication system at every level and in every part of the system, an intelligent cyber-physical system is proposed. As stated in the European Commission's Horizon 2020 funding programme [8], there is a strong requirement for smart cyber-physical systems. An intelligent cyber-physical system should have, as defined in the Industry 4.0 mission, besides the existence of computing element, an intelligent control system, capable of changing not only the system's behaviour but also the functional architecture and operational structure. In the context of a cyber-physical system incorporating a manufacturing system, this may mean, as per the change in the user's requirements the system may change the process flow, change which machine will be used and which will be bypassed, all through an intelligent decision making system. Such a functionality and system architecture would bring a new dimension to the cyber-physical systems. It would increase the scalability and reusability of a system, bringing in agility and fast adaptability with decreased cost and time – critical factors in today's highly competitive markets and fast

Received: January 2019, Accepted: June 2019
Correspondence to: Dr Vaibhav Shah
Department of Information Systems, Azurém Campus,
University of Minho, 4804-533 Guimarães, Portugal
E-mail: shah@dsi.uminho.pt
doi:10.5937/fmet1904802S

changing requirements. While there are examples of manufacturing systems integrated with digital and networked layers for remote operations [9], the functionality of fast adaptability requires an intelligent control system. This intelligent control system is achieved through a machine learning algorithm embedded in the system, with the ability to be trained and retrained at any point in time, hence the ability to change the process flow and system behaviour with little efforts in terms of physically changing the configuration, and utilizing the maximum potential of computational capacities and sophisticated techniques. This new paradigm of intelligent cyber-physical systems is emerging fast and is expected to transform the industry and infrastructure faster than the IT revolution in the past decades [10].

This paper presents an architecture for an intelligent cyber-physical system with the aforementioned capabilities, i.e. a cyber-physical system, with manufacturing system as part of its physical structure/system and a control system with empowered by machine learning algorithm for training the system and for real-time decision making to operate the manufacturing system. The machine learning algorithm used for the proposed system is a supervised learning algorithm for grammatical inference, which is an inductive inference based learning algorithm.

The rest of the paper is organised in the following sections. Section 2 presents the proposed cyber-physical system's architecture. In this section, first in subsection 2.1 a manufacturing system is explained with its main components and the corresponding process flow, to highlight the control system and its function. Then in subsection 2.2 the proposed system's architecture is presented with the integrated machine learning module. In section 3, the applied grammatical inference based machine learning algorithm is explained, along with the data preparation process, to map the manufacturing processes into grammar symbols. In section 4, the grammar synthesis process, for the presented case, is explained along with the learning module's ability of re-learning and re-training the control system. The section 4 also presents experimental results as validation of the approach and proves the feasibility of the proposed I-CPS architecture with functionality of reconfiguration of manufacturing system and intelligent decision making support to the control unit. The concluding remarks are presented in Section 5.

2. MANUFACTURING SYSTEM CONTROL AND CYBER-PHYSICAL SYSTEM ARCHITECTURE

For the presented work, we have considered a manufacturing system presented in the previous work [11] as the model test case scenario. In the present work a different machine learning algorithm is applied. Also, in the presented work, the previous manufacturing system is presented as part of the larger multi-layered cyber-physical system architecture. The objective is to present an intelligent cyber-physical system architecture that has a production/manufacturing system as well as an intelligence generator module, enabled with decision making ability using machine learning algorithm, embedded in the design itself.

2.1 Manufacturing system components

Representing different types of manufacturing systems from various industries and contexts is difficult as they are so diversified in terms of configurations and components, but they must be designed considering their steady state and dynamic performance [12]. A manufacturing system could be thought of as a set of production machines, but as explained in [13] a truly integrated manufacturing system cannot be considered complete without some form of co-ordination and control systems. As defined more recently in [14], a manufacturing system is "a collection of integrated equipment and human resources, whose function is to perform one or more processing and/or assembly operations on a starting raw material, part, or set of parts". Summarising these and other works in the area [15, 16], a typical modern day computerised manufacturing system includes the following types of components: production line machines, transportation systems and one or more types of control systems. The production line machines could be either CNC (computer numerical control) machines, welding machines, a robotic arm or an assembly line system, or a combination of these. The transportation systems could be human based, robotic arm, a conveyer belt or an AGV (automated guided vehicle). The control system could be one or a combination of the following: human resources (operators etc.), PLC (programmable logic controller), a hybrid system of sensors or some other form of more sophisticated system. The Figure 1 shows a process flow diagram (PFD) of an example manufacturing system with production line, involving various processes, such as machining and transportation.

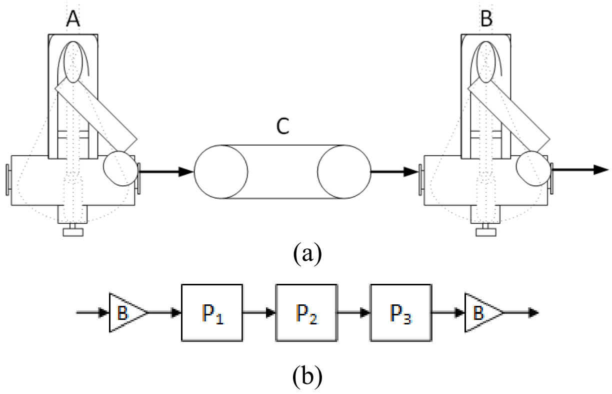

Figure 1. Process flow diagrams of a manufacturing system

The Figure 1(a) shows a process flow diagram of a manufacturing cell. This system has milling machines (A and B), with a conveyer belt (C) between them, transporting the pieces from one machine to the other, i.e. in the given diagram from A to B. A smooth running of a manufacturing system is taken care by buffering mechanism. Buffering makes sure that the production process continues despite unforeseen factors and change in the rate of supply and demand. Especially in the case of a cellular manufacturing system (CMS), such as the one in consideration here, buffer handling is critical [17]. The presented case is an example of serial production where 'buffer' is processed by FIFO – first in first out, order, i.e. the process that starts first, ends first, the piece that passed through the first job goes on to the next job. The second piece is considered for the current job only after the first piece is processed for the current job. Now, consider that each of the machines in Figure 1(a) have further sub-processes, represented by P_1, P_2 and P_3. Thus, each machine has an input from the buffer from the previous machine or from the buffer of the manufacturing cell, followed by the local sub-processes followed by the exit buffer. Such a process flow, on a single machine, is represented in the process flow diagram in Figure 1(b).

In the Figure 1(b), the entry buffer and exit buffer are denoted by B, while the three serial processes are P_1, P_2, and P3. Thus, a state sequence for jobs on a piece on a given machine would be Buffer > Process P_1 > Process P_2 > Process P_3 > Buffer. These sequences are presented in the Figure 2.

Figure 2 shows the entry and exit buffers as well as sub-processes on each of the machines in this manufacturing cell configuration. Here, it is important to note that the process flow sequence is managed by the 'control' unit of the cell. In the traditional manufacturing cell, this control unit is a simple computer controller or a human-based controller. However, the presented work proposes a machine learning based system that interacts with the cell through its control system.

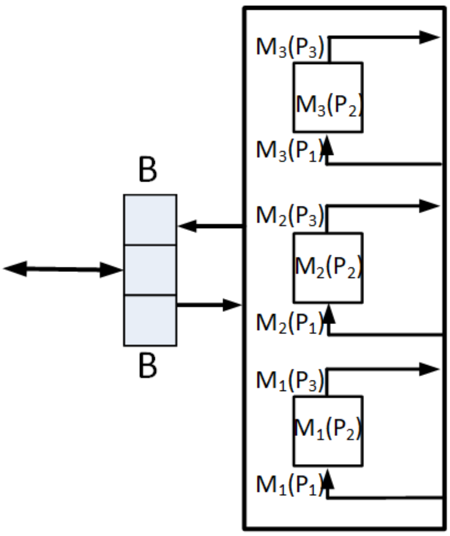

Figure 2. Process flow diagram with machine processes

The supposed process flow of the system in Figure 2 is such that the system interacts with the buffer to receive and supply pieces. Once a piece enters into the system through the buffer (B), it may go to either machine 1, or machine 2 or machine 3 (denoted by M_1, M_2 and M_3, respectively), or it may go through multiple machines, but in that order/direction only, before moving on to buffer B outside the system. Thus, for example if a piece is passed from B to M_2, it may go back to B after finishing the processes, or go to M_3 for further processes and then go to B. But it may not go from M_2 to M_1. Hence, the valid process flows examples are as given in Figure 3.

In the examples in Figure 3, the symbol '→' implies a transition from the left to right. The process must start and end with the buffer. It only goes in forwarding direction. A machine may or may not become a part of a given execution. This flow is the basic nature of our case, and the user may want to allow only some of these paths to be executed in the system, while restricting the system of other paths. In the sections 3 and 4 the grammar synthesis process, to support the desired system behaviour, is described.

$B \rightarrow M_1 \rightarrow B$
$B \rightarrow M_2 \rightarrow B$
$B \rightarrow M_3 \rightarrow B$
$B \rightarrow M_1 \rightarrow M_2 \rightarrow B$
$B \rightarrow M_1 \rightarrow M_3 \rightarrow B$
$B \rightarrow M_2 \rightarrow M_3 \rightarrow B$
$B \rightarrow M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow B$
$B \rightarrow M_1 \rightarrow B \rightarrow B \rightarrow M_2 \rightarrow B$
$B \rightarrow M_3 \rightarrow B \rightarrow B \rightarrow M_2 \rightarrow B$
...

Figure 3. Valid examples of process flow of the presented manufacturing system

2.2 Intelligent cyber-physical system architecture

After presenting our manufacturing cell, we now present the architecture of the proposed intelligent cyber-physical system, as shown in Figure 4. This architecture is adapted from the architecture presented in [18]. This subsection gives a brief description of the proposed system architecture.

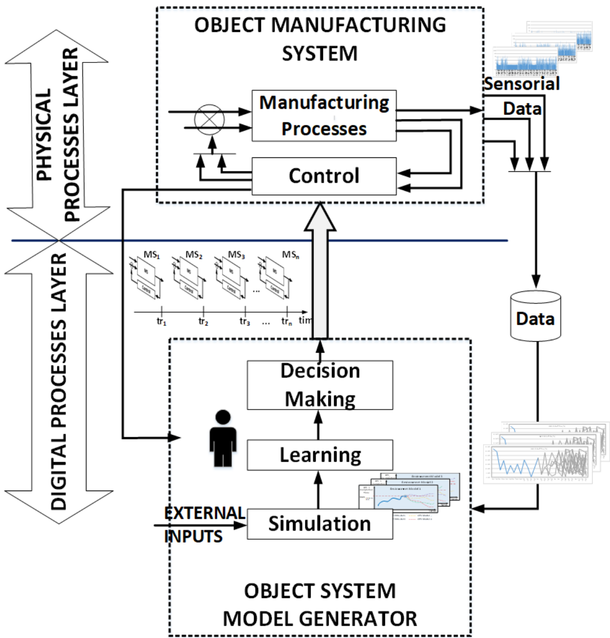

Figure 4. Intelligent cyber-physical manufacturing System architecture with adaptation from [18]

In the adapted architecture, the cyber-physical system is represented in the form of layers. As mentioned earlier, a CPS has the software/digital part, i.e. computational, communicational and controlling part(s) present in every physical part. Hence, instead of showing hardware (physical) and software (digital) as two distinct parts of the system, they are presented as two layers of the system and these two layers are present in each part of the system. Thus, there is a 'physical processes layer' and a 'digital processes layer' as seen in Figure 4. The previously described manufacturing cell is denoted as 'Object Manufacturing System' (OMS) in the Figure 4. The OMS includes the manufacturing processes as well as the control system(s) unit, assembled and configured to manufacture the object (the product). The first entry to and final exit from the control systems are handled by the buffering system as discussed earlier. The control systems work in a feedback loop wherein the output from one cycle of manufacturing processes is passed on as the input for the next cycle of the manufacturing processes. Between two consecutive cycles, the control systems verify and decide the next machine(s) to include in the process flow and send the commands to. After each execution of the OMS, the sensorial data is gathered. This data is a report of which machines and processes were part of the actual processes that occurred, as well as data from their sensors. This data is compiled and passed on to 'Object System Model Generator' (OSMG) which functions on the digital processes layer. The data from the OMS may help in determining whether the machines functioned as planned or whether there were any anomalies in terms of processing time or other parameters of machines. This data, along with other external inputs, such as change in production planning, eliminating some processes or introducing new processes, are compiled to run the simulator in the OSMG, and new examples are generated to train the learning module. As a result, a model is generated, which provided to the decision making module. This block, existing on the digital processes layer, is the intelligent module that creates the model of the system and provides assistance to the OMS. Here, the manufacturing system itself becomes the object whose model is generated through machine learning enabled intelligent module. Hence, it is called the object system model generator.

During the learning phase a human oracle is involved to decide which routines should be sent as examples for the next product planning. Although this process can be automatized as well, the purpose is not to eliminate the human role.

3. MANUFACTURING SYSTEM MODEL SYNTHESIS

For the presented case, we have found the inductive inference learning approach to be more appropriate, due to its ability to quickly synthesise a generalized model based on codified examples of the process. The objective is to generate a formal grammar as a model of the production line process in the given context. Below, the formal grammar is described as a 4-tuple:

Grammar G = (V, N, S, P).

where,

V is a finite set of terminal symbols,

N is a finite set of non-terminal symbols (states),

S is the start symbol (also called the sentence symbol), and

P is a finite set of production rules.

These rules define the transitions from one state to another for any input, from the set V.

In the following sub-sections, the algorithm outline, tools for performing the training and testing tasks as well as the model (grammar) synthesis for the presented case, i.e. how the learning algorithm is applied in this case run and how to interpret its output are described.

3.1 Model synthesis algorithm

The regular grammar was considered as suitable representation model due to the linear nature of the sequence. Successor Method algorithm was selected for the grammar inference task, i.e. to synthesise manufacturing system model synthesis. This learning algorithm was originally developed by two different teams [19], [20] and subsequently also presented in [21]. The Successor Method algorithm only takes positive examples into account. Based on these examples, a Regular Grammar is generated which can easily distinguish between 'positive' and 'negative' examples. In our case, a 'positive example' is a sequence of symbols that represents the manufacturing system's process flow correctly, as defined by the user. A wrong or invalid sequence of operations is clearly a 'negative example.' The outline of the learning algorithm is provided in Figure 5.

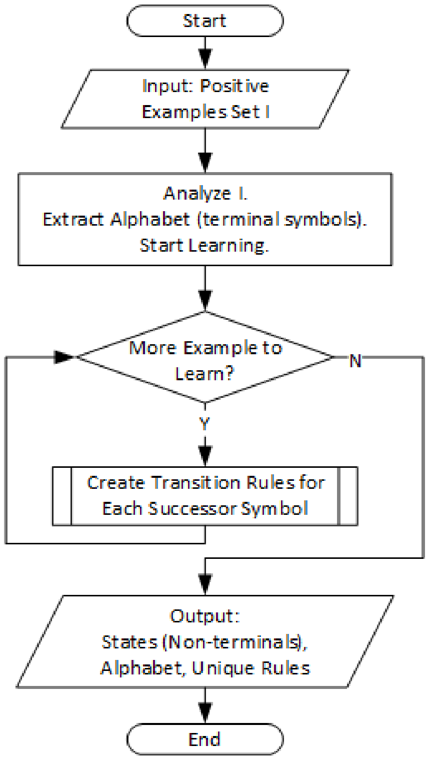

Figure 5. Successor method learning algorithm outline

The learning algorithm takes one positive example at a time, provided in the form of strings of symbols, called terminal symbols, and starts processing the strings from left to right, symbol by symbol. At the beginning of each input example (string), the system is believed to be at the Start state, and from there on, for each successor symbol, a transition rule is generated, considering the preceding symbol as input to the current state and succeeding symbol as the output, along with a non-terminal symbol (a state). In other words, the first symbol of the string implies the system moves from the Start state with that input. Similarly, the last symbol implies that the system stops when that symbol is received as input.

The inductive inference from positive examples means deriving general rules by learning from examples that correctly represent a system. The inferred rules represent the model of the system, and this model should be able to identify and generate positive examples that were not used during the training, i.e. the ability to learn a model that can actually generate new information. Furthermore, these rules should not satisfy a negative example while presented in the testing mode.

3.2 Data preparation for grammatical inference

In this sub-section it is explained how the data is prepared to send to the machine learning module in order to generate a grammar (model) of the manufacturing system. The input data is prepared from the elements of the manufacturing system presented in Figures 1 and 2.

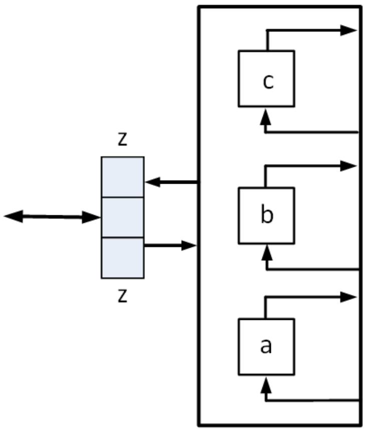

Figure 6. Revised process flow diagram with grammar symbols

As explained in section 3.1, the algorithm accepts strings of symbols and derives grammar rules. Hence, the first step is to represent the manufacturing system in terms of symbols. By the accepted norms of grammatical inference, these symbols are lowercase symbols, called the terminal symbols (as they form the final strings generated by the grammar) [22]. Consider the process flow diagram in the Figure 2, again. As seen in the diagram, there are always three step sub-processes for each machine that happen in the same order. For example, on machine M_1 the processes P_1, P_2 and P_3 always happen in that order, hence the whole of machine M_1's processes can be represented by a single symbol, say 'a'. Similarly, the processes of machines M_2 and M_3 can be described by symbols 'b' and 'c' respectively. Finally, the buffer is always handled by the same block, hence it can be denoted by a single symbol, let's call it 'z'. With this new symbolism, the process flow diagram in the Figure 2 can be redrawn as shown in Figure 6.

Thus, if a product manufacturing routine is to start from machine M_1, bypass machine M_2 and move to M_3 before exiting the cycle, the whole process can simply be denoted, by this new system as 'zacz' only. Thus, the processes in the Figure 3 can be rewritten as strings shown in Table 1.

In the next section, the grammar synthesis process and the model regeneration process for system reconfiguration are described.

Table 1. String equivalents of manufacturing processes

Processes	String equivalent
$B \rightarrow M_1 \rightarrow B$	zaz
$B \rightarrow M_2 \rightarrow B$	zbz
$B \rightarrow M_3 \rightarrow B$	zcz
$B \rightarrow M_1 \rightarrow M_2 \rightarrow B$	zabz
$B \rightarrow M_1 \rightarrow M_3 \rightarrow B$	zacz
$B \rightarrow M_2 \rightarrow M_3 \rightarrow B$	zbcz
$B \rightarrow M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow B$	zabcz
$B \rightarrow M_1 \rightarrow B \rightarrow B \rightarrow M_2 \rightarrow B$	zazzbz
$B \rightarrow M_3 \rightarrow B \rightarrow B \rightarrow M_2 \rightarrow B$	zczzbz

4. MODEL GENERATION THROUGH INDUCTIVE INFERENCE

For the grammar synthesis through inductive inference, a selected set of training examples is passed to the algorithm. The algorithm is implemented as a software tool which has been presented in a previous work [shah]. The examples are selected in order to generate a grammar that allows certain production routines (process flows) while prohibits others. In the sub-section 4.1, first the grammar synthesis process is shown using a set of examples, and in sub-section 4.2 a new grammar is generated by changing the training set. The grammar is the model of manufacturing system which is sent to the control system (Figure 4). By changing the training examples set, the system model is changed, and this changes the behaviour and outcome of the manufacturing system.

4.1 Grammar synthesis

For the first experiment, the training set I_1 was defined as follows:

I_1 = {zz, zaz, zabz, zabcz}

As can been seen in the training set, all the examples used show the process flows that start at 'a' immediately after the buffer. Hence, this is the desired behaviour by the user, for the first production setup.

The grammar synthesis process, from a training set, using the successor method, has been explained in [23], hence for the purpose of the current work, the result of the synthesis process is directly presented. Applying the successor method algorithm to the above training set, using the grammar synthesis tool, the regular grammar G_1 was synthesised as shown in Figure 7.

The grammar G_1 can be represented as 4-tuple,

$G_1 = (V, N, S, P)$

Where,

$V = \{z, a, b, c\}$,

$N = \{Z, A, B, C\}$,

$S = \{S\}$,

$P = \{S \rightarrow Z, Z \rightarrow z, Z \rightarrow zA, Z \rightarrow zZ, A \rightarrow aB, A \rightarrow aZ, B \rightarrow bC, B \rightarrow bZ, C \rightarrow cZ\}$.

The grammar G_1 was trained using 4 positive examples. However, the grammar can actually parse many more positive examples that were not used for the training. At the same time, it rejects negative examples. Here, parsing means approving an input string when validated through the grammar. The Figure 8 and Table 2 show actual parser screen and detailed parsing results, respectively. Here the terms positive and negative examples refer to the production routines that are desired and not desired, respectively, by the user or the system designer and controller.

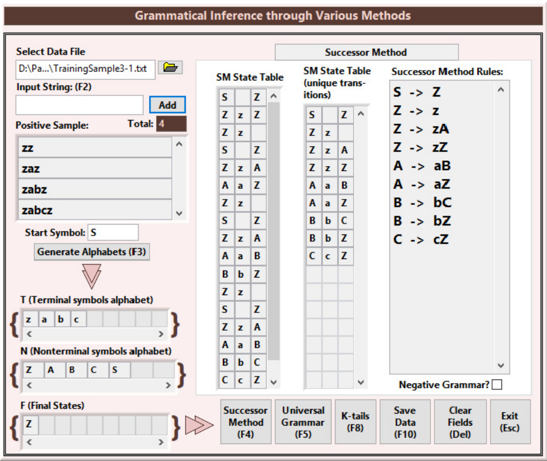

Figure 7. Synthesis of regular grammar G1 by successor method [23]

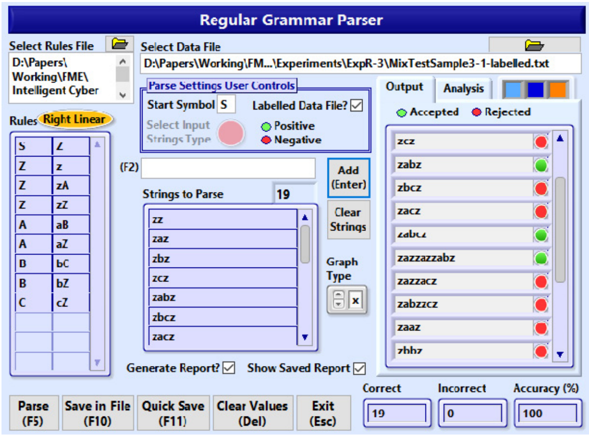

Figure 8. Results by the specially developed parser tool

Table 2. Results of parsing by grammar G1 (100% correct)

Examples	Expected	Result
zz	Accept	Accepted
zaz	Accept	Accepted
zbz	Reject	Rejected
zcz	Reject	Rejected
zabz	Accept	Accepted
zbcz	Reject	Rejected
zacz	Reject	Rejected
zabcz	Accept	Accepted
zazzazzabz	Accept	Accepted
zazzacz	Reject	Rejected
zabzzcz	Reject	Rejected
zaaz	Reject	Rejected
zbbz	Reject	Rejected
zccz	Reject	Rejected
zaabz	Reject	Rejected
zacbz	Reject	Rejected
zbcaz	Reject	Rejected
azabcz	Reject	Rejected

Although the actual experiments were done using many more examples, for the sake of results analysis, only a selected number of example are presented. Consider the two examples 'zazzazzabz' and 'zazzabz' that were not used in the training, but show the desired behaviour, i.e. the control must send the process from buffer to 'a' only, and from there it may go to other machines or back to buffer 'z'.

Now, suppose, the user wants to change the system's behaviour and wants that the process may start from either of 'a' or 'b' or 'c' and continue further in that direction.

4.2 System reconfiguration

To reconfigure the system, in the proposed architecture, the user does not need to change the physical layout of the manufacturing cell, nor does the user need to change the physical configuration of the CPS. The proposed I-CPS can be reconfigured by supplying a different set of training examples.

Consider the new training set I_2.

I_2 = {zz, zaz, zbz, zcz, zacz, zabcz}

The revised training set has 3 new examples, 'zbz', 'zcz' and 'zacz' while the example 'zabz' has been removed because it became redundant, meaning its existence did not change the output grammar.

The grammar synthesised from the training set I_2 is G_2 as following.

$G_2 = (V, N, S, P)$

Where,

V = {z, a, b, c},

N = {Z, A, B, C},

S = {S},

P = {S $\rightarrow$ Z, Z $\rightarrow$ zA, Z $\rightarrow$ z, Z $\rightarrow$ zC, Z $\rightarrow$ zB, Z $\rightarrow$ zZ, A $\rightarrow$ aC, A $\rightarrow$ aZ, A $\rightarrow$ aB, C $\rightarrow$ cZ, B $\rightarrow$ bC, B $\rightarrow$ bZ}.

For the sake of comparison, the same set of examples were parsed again through this grammar. In this case, several of the previously invalid process routines were now valid. The Figure 9 shows the actual parser output screen, while the Table 3 shows the results of all the test examples. It is worth noticing that the results are again 100% accurate, which means the control system will only allow the desired behaviour in the manufacturing system.

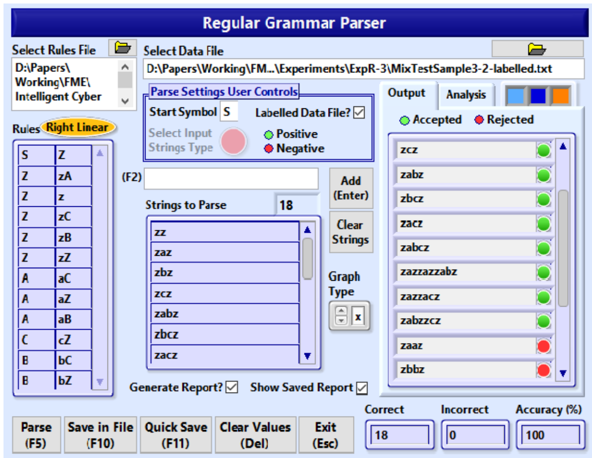

Figure 9. Parsing by grammar G_2

Table 3. Results of parsing by grammar G_2 (100% correct)

Examples	Expected	Result
zz	Accept	Accepted
zaz	Accept	Accepted
zbz	Accept	Accepted
zcz	Accept	Accepted
zabz	Accept	Accepted
zbcz	Accept	Accepted
zacz	Accept	Accepted
zabcz	Accept	Accepted
zazzazzabz	Accept	Accepted
zazzacz	Accept	Accepted
zabzzcz	Accept	Accepted
zaaz	Reject	Rejected
zbbz	Reject	Rejected
zccz	Reject	Rejected
zaabz	Reject	Rejected
zacbz	Reject	Rejected
zbcaz	Reject	Rejected
azabcz	Reject	Rejected

Again, as visible in the results, all the valid strings are accepted by the grammar, while all the invalid strings are rejected.

4.3 Results interpretation

The grammar synthesis is the process where the user supplies a representative set of examples to the machine learning module, in the OSMG, and the module generates a model, which is supplied to the control system via 'Decision Making' module. This means the control system can now command the manufacturing system only to behave by the process flows which are generated by 'phrases' that the supplied model (grammar) can generate. The grammar G_1 can only generate phrases such as 'zazzabz' (Buffer → M_1 → Buffer → Buffer → M_1 → M_2 → Buffer) and so on. This was also the desired behaviour for the user. The user, according to the requirement, may use the generated phrase or not for a particular product. The OSMG produced process flows like this one even though the learning module was not supplied by the user. This capacity is achieved by the power of the inductive inference based learning module.

Similarly, the parser function works at the end of each cycle, to analyse if the machine behaved according the grammar approved by the user. Also, the parser functions as a controller to check whether the product specifications supplied by the user are in accordance with the model currently in use. Hence, the system provides assistance to the control system at multiple stages. It also helps in the form of sending the commands to the manufacturing system as well as checking whether the commands were executed correctly or not.

5. CONCLUSION

An architecture for intelligent cyber-physical system (I-CPS), enabled by machine learning algorithm with very high accuracy, is proposed. The learning module provides the ability to reconfigure the manufacturing system without great effort or physically changing the system configuration for each new product process flow. The intelligent cyber-physical system, although capable of functioning autonomously, has a human at its centre for the command and control and to decide which examples to use to teach (train) the system and periodically check the system performance and as well as its behaviour. Moreover, the system can generate and accurately check process flow examples that were previously not supplied by the user. This is the learning achieved in its true sense. The proposed system architecture has all the important features desired in an intelligent cyber-physical system, capable of decision making in real-time and changing its behaviour accordingly. This is in accordance with the Industry 4.0.

ACKNOWLEDGMENTS

This work has been supported by FCT – Fundação para a Ciência e Tecnologia, Portugal, within the Project Scope: UID/CEC/00319/2019

REFERENCES

[1] Ferreira, L., Putnik, G., Cruz-Cunha, M. M., Putnik, Z., Castro, H., Alves, C., Shah, V., Varela, M. L. R.: Cloudlet Architecture for Dashboard in Cloud and Ubiquitous Manufacturing, Procedia CIRP, Vol. 12, pp. 366-371, 2013.

[2] Kagermann, H., Lukas, WD.: Industry 4.0: With the Internet of Things on the way to the 4th Industrial Revolution, in Technology & Society, Issue 13, 2011.

[3] Bortolini, M., Galizia, F. G., Mora, C.: Reconfigurable manufacturing systems: Literature review and research trend, Journal of Manufacturing Systems, Vol. 49, pp. 93-106, 2018. DOI: 10.1016/j.jmsy.2018.09.005.

[4] Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., et al.: Reconfigurable manufacturing systems, CIRP annals— manufacturing technology, Vol. 48, No. 2, pp. 527–40, 1999.

[5] Esmaeilian, B., Behdad, S., Wang, B.: The evolution and future of manufacturing: a review, Journal of Manufacturing Systems, Vol. 39, pp. 79–100, 2016.

[6] Gill, H.: From Vision to Reality: Cyber-Physical Systems, HCSS National Workshop on New Research Directions for High Confidence Transportation CPS: Automotive, Aviation, and Rail November 18-20, 2008.

[7] Lee, E. A.: The Past, Present and Future of Cyber-Physical Systems: A Focus on Models, Sensors, Vol. 15, no. 3, pp. 4837-4869, 2015.

[8] Horizon 2020: Smart Cyber-physical Systems, https://ec.europa.eu/programmes/horizon2020/en/h2020-section/smart-cyber-physical-systems

[9] Brkić, V. S., Putnik, G., Shah, V., Castro, H., Veljković, Z.: Human - Computer Interactions and

User Interfaces for Remote Control of Manufacturing Systems, FME Transactions, Vol. 41, pp. 250-255, 2013.

[10] Müller, H. A.: The Rise of Intelligent Cyber-Physical Systems, Computer - Computing: The Next 50 Years, IEEE, 2017.

[11] Putnik, G. D., Rosas, J. A.: Manufacturing System Design: Towards Application of Inductive Inference, in *Proceedings of the 3rd International Workshop on Emergent Synthesis – IWES'01*, 12-13.03.2001, Bled, Slovenia, pp. 171-180.

[12] Parnaby, J.: Concept of a manufacturing system, International Journal of Production Research, Vol. 17, No.2, pp. 123-135, 1979. DOI: 10.1080/00207547908919600.

[13] Collins, J.: Integrated manufacturing--the state of the art, Production Engineer, Vol.59, No.6, pp. 41-44, 1980. DOI: 10.1049/tpe:19800099.

[14] Groover, M. P.: *Automation, Production Systems, and Computer-Integrated Manufacturing*, Third Edition, Pearson Education Inc, 2008.

[15] Putnik, G. et al.: Scalability in manufacturing systems design and operation: state-of-the-art and future developments roadmap, CIRP Ann, Vol. 62, No.2, pp.751–774, 2013.

[16] ElMaraghy, H., Caggiano, A.: Flexible Manufacturing System, in: Laperrière, L., Reinhart, G. (eds): *The International Academy for Produ, CIRP Encyclopedia of Production Engineering*. Springer, Berlin, Heidelberg, 2016.

[17] Wijngaard, J.: The effect of interstage buffer storage on the output of two unreliable production units in series, with different production rates, AIIE Trans., Vol.11, pp. 42-47, 1979.

[18] Putnik, G. D., Ferreira, L., Lopes, N., Putnik, Z.: What is Cyber-Physical System: Definitions and Models Spectrum, FME Transactions, Vol. 47 No. 4, pp. 663-674, 2019.

[19] Richetin, M., Vernadat, F.: Efficient regular grammatical inference for pattern recognition, Pattern Recognition, Vol. 17, No. 2, pp. 245-250, 1984.

[20] Rodger R.S., and Rosebrugh, R.D.: Computing a grammar for sequences of behavioural acts, Animal Behaviour, Vol. 27, pp. 737-749, 1979.

[21] Miclet, L.: Structural Methods in Pattern Recognition, Chapman and Hall, 1986.

[22] Chomsky, N.: On Certain Formal Properties of Grammars, Information and Control, Vol. 2, No.2, pp. 137-167, Elsevier, 1959. DOI: 10.1016/S0019-9958(59)90362-6

[23] Shah, V., Putnik, G. D.: Software Tools for Understanding Grammatical Inference Algorithms: Part I – Tools for Regular Grammars and Finite-State Automata, FME Transactions, Vol. 44, pp. 83-91, 2016.

СИСТЕМ КОНТРОЛЕ ПРОИЗВОДЊЕ НА ОСНОВУ МАШИНСКОГ УЧЕЊА ЗА ИНТЕЛИГЕНТНЕ САЈБЕР-ФИЗИЧКИ СИСТЕМИ (И-СФС)

В. Шах, Г.Д. Путник

Сајбер-Физички Системи (СФС) су често погрешно схваћени као било који уграђени систем. Прави сајбер-физички систем треба да има и физичке и дигиталне (рачунско-комуникационе-контролне) делове међусобно повезане у сваком делу и процесу, а сам систем треба да има способност да мења своје понашање да би се прилагодио променљивим захтевима.

Овај рад представља архитектуру Интелигентног Сајбер-Физичког Система (И-СФС) где је реконфигурабилни производни систем подржан алгоритмом машинског учења који обезбеђује унапређено доношење одлука у систему контроле производње. Приказани су резултати експеримената који показују да се модул за машинско учење може помоћи да се систем контроле сам прилагоди промнама у спољним захтвима (од стране корисника) у облику примера за обуку. Резултат је архитектура интелигентног сајбер-физичког система, са физичким и дигиталним деловима који увек раде у синхронизацији, омогућавајући промену у понашању система у смислу тока производног процеса како би се прилагодили свакој промени у планирању производње.