INFORMATICA, 2021, Vol. 32, No. 2, 217–245
© 2021 *Vilnius University*
DOI: https://doi.org/10.15388/20-INFOR441

Controlling Supervised Industry 4.0 Processes through Logic Rules and Tensor Deformation Functions

Borja BORDEL[1,*], Ramón ALCARRIA[2], Tomás ROBLES[1]

[1] *Department of Information Systems, Universidad Politécnica de Madrid, Spain*
[2] *Department of Geospatial Engineering, Universidad Politécnica de Madrid, Spain*
e-mail: borja.bordel@upm.es, ramon.alcarria@upm.es, tomas.robles@upm.es

Received: July 2020; accepted: December 2020

Abstract. Industry 4.0 solutions are composed of autonomous engineered systems where heterogeneous agents act in a choreographed manner to create complex workflows. Agents work at low-level in a flexible and independent manner, and their actions and behaviour may be sparsely manipulated. Besides, agents such as humans tend to show a very dynamic behaviour and processes may be executed in a very anarchic, but correct way. Thus, innovative, and more flexible control techniques are required. In this work, supervisory control techniques are employed to guarantee a correct execution of distributed and choreographed processes in Industry 4.0 scenarios. At prosumer level, processes are represented using soft models where logic rules and deformation indicators are used to analyse the correctness of executions. These logic rules are verified using specific engines at business level. These engines are fed with deformation metrics obtained through tensor deformation functions at production level. To apply deformation functions, processes are represented as discrete flexible solids in a phase space, under external forces representing the variations in every task's inputs. The proposed solution presents two main novelties and original contributions. On the one hand, the innovative use of soft models and deformation indicators allows the implementation of this control solution not only in traditional industrial scenarios where rigid procedures are followed, but also in other future engineered applications. On the other hand, the original integration of logic rules and events makes possible to control any kind of device, including those which do not have an explicit control plane or interface. Finally, to evaluate the performance of the proposed solution, an experimental validation using a real pervasive computing infrastructure is carried out.

1. Introduction

Industry 4.0 (Lasi *et al.*, 2014) refers to a new industrial revolution where current production solutions and robots are being replaced by Cyber-Physical Systems (Bordel *et al.*, 2017b) and other innovative engineered systems such as pervasive computing or sensing infrastructures (Ebling and Want, 2017). Traditionally, in industrial scenarios, managers

*Corresponding author.

define at a very high abstraction level processes to be executed and supported by production systems (Sánchez *et al.*, 2016). Dashboards and other graphic environments are employed for this purpose. In these models, actions to be performed at low-level together with their input parameters are indicated. Connections among tasks and their temporal organization are strict and cannot be modified. Moreover, sometimes valid ranges for task outputs may also be defined as common industrial systems and robots are very predictable and precise (Bordel *et al.*, 2018c). This top-down approach employs technologies such as YAWL (Van Der Aalst and Ter Hofstede, 2005) or BPMN (Geiger *et al.*, 2018) to create and validate processes and assumes that every low-level infrastructure is following a request-actuation design paradigm. In this paradigm, every low-level system is provided with an interface through which requests may be received. After each request, the hardware system performs a certain action or actuation and, after that, stops its operation waiting for a new request (Bordel *et al.*, 2018b). These systems, then, are totally controllable by external agents and match perfectly the request sequences that are created through modelling technologies such as YAWL (Bordel *et al.*, 2017a).

However, Industry 4.0 scenarios are different. Cyber-Physical Systems and pervasive computing infrastructures are not typically provided with open interfaces, and they tend to act autonomously according to deterministic algorithms or, even, learning technologies (Bordel *et al.*, 2020b). Dense environments where thousands of agents with heterogeneous capabilities are deployed and working together are the most common application scenarios for Industry 4.0. This context, furthermore, can get more complex if humans are considered (Bordel *et al.*, 2017c). In fact, Industry 4.0 is inclusive, and many manufacturing companies produce handmade products where human labour is essential. Production systems including humans are even more heterogenous, as people's behaviour is very variable and dynamic. And, obviously, human agents are not externally controllable (Bordel *et al.*, 2017c). Thus, in Industry 4.0, complex tasks and services are provided through the choreographed coordination of heterogenous agents acting in an autonomous way (Bordel *et al.*, 2018a). This bottom-up approach is also compatible with computational processes defined at high-level, if decomposition and transformation engines are considered. However, it is a very costly and inefficient approach, as many false negative alarms are triggered. When autonomous agents are included, processes may be executed in a very anarchic but still correct manner.

Therefore, traditional industrial control mechanisms are not valid for these scenarios, and innovative technologies are needed (Bordel *et al.*, 2020a). Then, in this work we propose a new supervisory control mechanism for Industry 4.0 scenarios, matching the special characteristics of distributed processes supported by coordinated autonomous and heterogeneous agents. This new technology defines (at prosumer level) processes using soft models where flexible logic rules and general deformation metrics guarantee the correctness of executions. These rules are verified in a passive manner at business level using specific engines. These engines receive information (events) from lower layers and run a validation procedure to analyse if minimum rules are being met by low-level agents. These events are enriched with information at production level, describing if tasks are being correctly executed or not. Finally, at production level, the physical parameters or tasks under execution are being monitored. Using tensor deformation functions (Bordel *et al.*, 2017a), the global similarity between the expected executions and the real actions is measured. To allow these calculations, processes are represented as discrete flexible solids in a multidimensional phase space. This generalized approach is focused on reducing the false negative alarms observed in traditional control systems. All information is acquired through observation and recognition mechanisms, which are not described in this paper, but already existing (Bordel *et al.*, 2019).

The rest of the paper is organized as follows: Section 2 introduces the state of the art on control mechanisms for Industry 4.0 scenarios. Section 3 presents the proposed technology, including the three abstraction levels (prosumer, business, and production) and their associated mechanisms. Section 4 provides an experimental validation of the proposal. Finally, Sections 5 and 6 explain some results of this experimental validation and the conclusions of our work.

2. State of the Art

Industry 4.0 is one of the most popular research topics nowadays (Lu, 2017), thus, many different control solutions for these new scenarios have been reported. In fact, almost every existing control mechanism has been already applied and integrated into Industry 4.0 technologies and scenarios: from traditional monolithic instruments (Kretschmer *et al.*, 2016) to most modern intelligent algorithms (Meissner *et al.*, 2017). Besides, a large catalogue of specific control solutions for Industry 4.0 scenarios have been also reported (Dolgui *et al.*, 2019).

Two basic types of control solutions for Industry 4.0 have been described: supervisory control (Wonham and Cai, 2019) and embedded control (Aminifar, 2016) solutions. In supervisory control mechanisms, a surveillance system monitors the behaviour of hardware infrastructures and actuates and intervenes in the production process if it goes outside an acceptable variation margin. On the other hand, embedded control mechanisms are integrated into the production processes themselves and are continuously regulating the evolution and behaviour of the hardware platform.

One of the most popular supervisory control mechanisms in Industry 4.0 are SCADA systems (Mohammad *et al.*, 2019) (Supervisory Control And Data Acquisition). SCADA solutions can manage heterogeneous infrastructures, through complex and heavy modular software tools (Calderón Godoy and González Pérez, 2018). Traditionally, SCADA solutions were built as monolithic platforms where specific industrial protocols, such as OPC, were employed (Boyer, 2016). Initial applications for Industry 4.0 also followed this paradigm (Merchan *et al.*, 2018). Nevertheless, recently, new and different modules for slightly distributed SCADA solutions have been reported (Branger and Pang, 2015), and even cloud-based platforms may be found (Sajid *et al.*, 2016). Furthermore, some SCADA mechanisms for industrial scenarios based on Internet-of-Things have been described (Wollschlaeger *et al.*, 2017). The main problem of SCADA systems is their security weaknesses: many different reports about security problems of SCADA systems in the context of Industry 4.0 scenarios have been reported (Igure *et al.*, 2006; Chhetri *et al.*, 2017). Moreover, in largely distributed production systems, communication delays usually create complex malfunctions in SCADA control functions. Then, different stability analyses (Foruzan *et al.*, 2016), mathematical models to compensate delays (Silva *et al.*, 2018) and evolution analyses (Gu *et al.*, 2019) to detect problems have been reported. In any case, these problems are still present and only distributed solutions including a small number of devices are working nowadays.

Other supervisory control solutions for future industrial scenarios based on autonomous agents have been described. These mechanisms are typically defined as Discrete event systems (Wonham *et al.*, 2018) and are focused on autonomous robots (Gonzalez *et al.*, 2018) and similar mobile machines (Roszkowska, 2002), although other applications to real-time solutions (Sampath *et al.*, 1995) and fault diagnosis (Moreira and Basilio, 2014) may be found. Logic rules have been also employed to implement robot navigation frameworks (Kloetzer and Mahulea, 2016) and different mechanisms to optimal (Fabre and Jezequel, 2009) or clean paths (Iqbal *et al.*, 2012) in robotized Industry 4.0 have been also reported. Works on this area are also evaluating, in a formal way, the scalability (Hill and Lafortune, 2017) and software characteristics (Goryca and Hill, 2013) of supervisory software solution for Industry 4.0.

Contrary to all these previous works, the problem and scenario addressed in this work is more general. First, we are considering not only robots and similar devices but also pervasive infrastructure and humans; what introduces an important challenge. Besides, all previous supervisory control solutions assume there is an interface so the surveillance system can intervene and act in the production process; however in most future engineered solutions (and, of course, when humans are considered), that's not a realistic assumption.

Embedded control solutions can be classified into two different groups: vertical and horizontal architectures (Dolgui *et al.*, 2019). Vertical architectures are, probably, the genuine approach for Industry 4.0 applications. In this approach, computational processes are transformed and decomposed (Ivanov *et al.*, 2016a; Bagheri *et al.*, 2015), so executable units may be transferred and delegated to remote production infrastructures or, even, cloud services (Bordel *et al.*, 2018c). On the contrary, horizontal architectures are traditional embedded control paradigms, which have been adapted to Industry 4.0 (Lalwani *et al.*, 2006). Feedback control systems are the most traditional approach. In these mechanisms, a complex production system is represented through a block diagram where different key indicators are calculated at each step (Disney *et al.*, 2006). Feedback loops guarantee that if any deviation is detected, that information is considered in previous steps to correct the situation. Feedback control solutions for traditional linear production schemes (Bensoussan *et al.*, 2009; Lin *et al.*, 2018) are very popular, but additional proposals for new non-linear schemes (Spiegler and Naim, 2017; Zhang *et al.*, 2019) may also be found. Furthermore, different studies about how randomness (Garcia *et al.*, 2012), disturbances (Scholz-Reiter *et al.*, 2011), and fluctuations (Yang and Fan, 2016) affect the global behaviour and performance of these feedback control mechanisms have been published. On the other hand, optimal control applications have been reported. Optimal control is the most common technique in horizontal control architectures for Industry 4.0 (Dolgui *et al.*, 2019).

In these scenarios, cloud production systems are the most common application for these technologies (Frazzon *et al.*, 2018; Rossit *et al.*, 2019). Optimal control is characterized by a process evolution that is not allowed to belong to certain states or areas in the phase state. With this view, solutions for optimal planning (Sokolov *et al.*, 2018) and efficient activity scheduling (Ivanov *et al.*, 2019) in Industry 4.0 may be found. As in previous topics, works on robustness and resilience analyses have also been reported (Aven, 2017; Ivanov *et al.*, 2016b).

The main problem of embedded control mechanisms in Industry 4.0 applications is that they require a total access to every component in the industrial system; as control modules must be integrated in every component and all of them must be interconnected to generate the global expected behaviour. Contrary to these systems, the proposed approach in this paper is also valid for proprietary solutions (very usual in industrial applications) which cannot be accessed or easily modified, as only a supervisory transversal component is able to support the whole control policy.

Finally, control mechanisms based on modern technologies such as artificial intelligence and fuzzy logic may be found (Diez-Olivan *et al.*, 2019). Although fuzzy logic is not a recent technology (Nguyen *et al.*, 2018), new solutions for industrial scenarios have been recently reported. These solutions are sparse, but some new proposals for nonlinear and event-driven processes may be found (Pan and Yang, 2017). In this context, case studies about real implementations are also a relevant contribution (Theorin *et al.*, 2017; Golob and Bratina, 2018). These technologies are very powerful but are not flexible enough to deal with anarchist executions caused by human behaviour, as learning algorithms need to previously observe every execution model to be accepted.

Table 1 shows in a systematized manner the main advantages and disadvantages, and differences in terms of the problem addressed, of previously described works.

3. New Supervisory Control Solution for Distributed Industry 4.0 Processes

This section describes the new proposal for supervisory control in Industry 4.0 scenarios, where processes are described using soft models, logic rules and deformation functions and metrics. Section 3.1 describes the global overview of the proposed solution. Section 3.2 presents the new soft models to represent processes at prosumer level. Section 3.3 analyses how logic rules (at business level) may be employed to verify in a flexible manner the process execution performed by autonomous agents and humans. And Section 3.4 describes proposed technologies for production level, where deformation functions and metrics are deployed to evaluate workflows and feed verification engines at business level.

3.1. *General Overview*

Figure 1 shows the proposed architecture for the described new control mechanism. At the highest level, managers are defining processes ① using any of the existing process description technologies, such as BPMN or YAWL. Industrial production processes are usually very large and complex, including many activities and tasks which may be partially related

Table 1
Systematized state of the art review.

State of the art proposals	Main advantages	Main disadvantages	Improvements in the proposed solution
Supervisory SCADA-based solutions.	They do not require computationally heavy algorithms to perform control.	Devices must be provided with a control interface. Stability problems appear in large and complex architectures.	Autonomous devices without control interface and humans can be controlled. A wider range of applications can be considered (not only classic industrial scenarios).
Discrete event systems.	Autonomous agents are supported.	Solutions are not flexible and only compatible with one kind of agent (robots, vehicles, etc.).	The solution can be applied to heterogenous scenarios where several types of agents are present.
Logic rules supported systems.	Autonomous agents are supported. Configurable control policies may be defined.	They only show a good performance with rigid typical industrial processes.	Deformation metrics and soft models allow controlling flexible and "anarchist" processes.
Horizontal embedded control solutions.	Complex systems can be easily controlled in a very precise manner.	Devices must be provided with a control interface fully functional. Proprietary closed solutions are not compatible with these schemes.	Autonomous devices without control interface and humans can be controlled. Compatible with all kind of devices. Any future engineered system is controlled without a transversal adaptation middleware
Vertical embedded control solutions	Complex and distributed processes can be controlled through several decomposition and transformation phases.		
Fuzzy logic and artificial intelligence-based control.	They can support very complex control schemes through powerful algorithms.	These solutions cannot be flexible and precise at the same time. Typically, they are specifically designed for individual applications.	The proposed solution can be deployed in a large catalogue of scenarios as it balances pattern recognition techniques with other more traditional approaches.

(or not). Besides, some technologies enable defining quality indicators for tasks' outputs, so the activity execution may be rejected if those indicators are not met. Process definitions at this level (named as prosumer level, as managers act as producers and consumers in the proposed technology) are typically graphic, and only relations among tasks and quality measurements are described.

The process model based on standard technologies (hereinafter we are considering YAWL language) is then analysed and decomposed ② to "soft" and relax the model. The idea is to transform a hard or rigid process definition where deformations and variations are not expressed (and, then, not accepted by the control solution) into a soft model ③ where global quality and organization restriction are equivalent but dynamic changes, variations and deformation are admissible.

To perform this transformation, a specific engine ② analyses the YAWL process model and identifies the key branches or subprocesses whose exterior structure cannot be modified. For example, a subprocess generating a subproduct as output which is the input of a second subprocess must be executed strictly before the second subprocess. However,

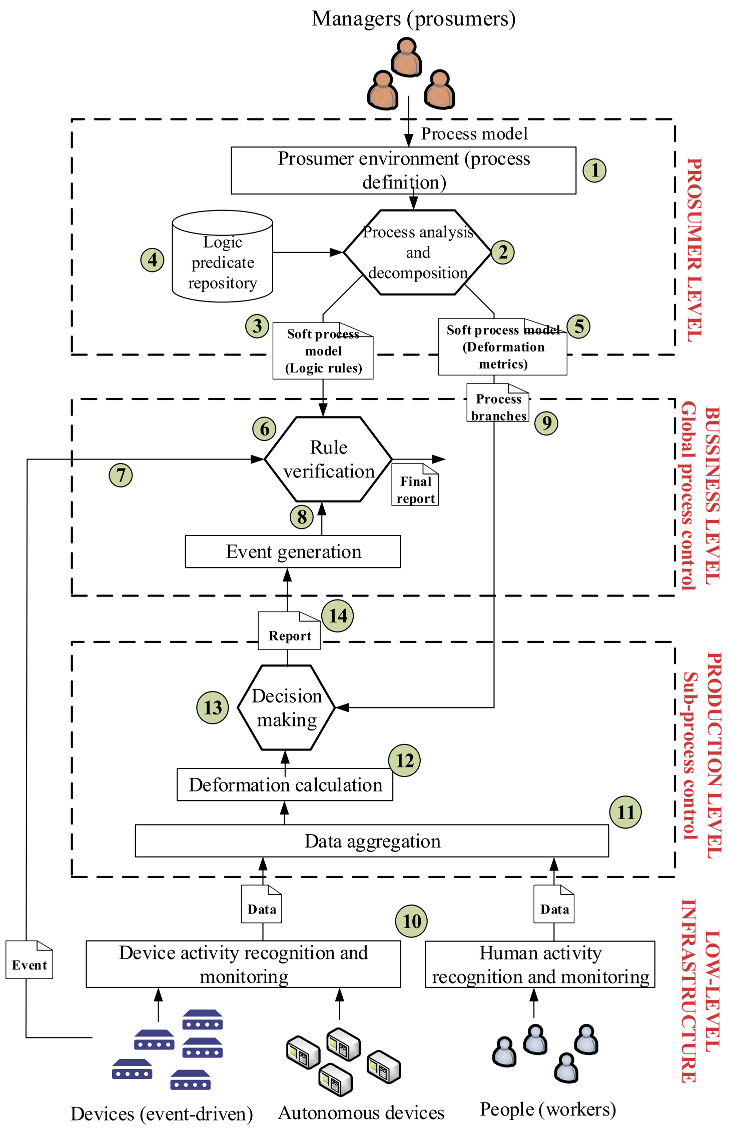

Fig. 1. Proposed architecture for new supervisory control solution in Industry 4.0 scenarios.

tasks inside both subprocesses could be executed in any order, and the same subproduct is created.

After this analysis and decomposition procedure, two different soft process descriptions are generated. The first one is a global description based on logic rules ③. To generate this description, each subprocess is represented by a discrete state. And the global state map is regulated by a set of logical rules extracted and instantiated from a logic predicate repository ④, where different general conditions are stored (temporal order, necessary conditions, etc.). Those logical rules represent the minimum restrictions that the production process must fulfill, but all additional restrictions artificially introduced by the limitations or graphic process representations are removed. The second one is a set of reports ⑤ where deformation metrics about each subprocess are indicated. Quality indicators about each task are considered together and three global metrics for the whole subprocess are obtained: stiffness, strength and ductility.

The global soft description is transferred to a verification engine ⑥, where logical rules are validated (or not) at real-time. This global understanding about the process takes place at business level, where information from managers (process models) and from autonomous hardware platforms are matched to determine if production processes are being correctly executed or not. This verification engine takes as input two information sources: events directly generated by hardware devices ⑦ following an event-driven paradigm; and events generated ⑧ by control component in lower layers (the production layer). These events are employed to validate logical rules and determine if minimum conditions to consider the process execution is valid are being met.

All reports about subprocesses ⑨ are transferred to a lower level (the production level), where low-level (physical) information is collected to evaluate if process deformation is above the proposed global metrics. To do that, information from different recognition technologies (human-oriented, device-oriented, etc.) is collected ⑩. These recognition technologies are not addressed in this paper, as any of the previously existing mechanisms may be integrated (Bordel *et al.*, 2019; Bordel and Alcarria, 2017). All information sources are integrated ⑪ to perform a global evaluation of process deformation. To obtain those global metrics a mathematical algorithm ⑫ is employed. This algorithm is deducted by understanding production processes as discrete flexible solids in a generalized phase space. Dynamic behaviour of autonomous devices is represented as external forces acting on the solid (process) and deforming it. Using different deformation functions (and the provided deformation metrics), it is evaluated if the final process execution is similar enough to the model to be considered as a valid execution or not. A decision-making engine ⑬ performs these functions. A report ⑭ and event-like result (whose format may be adapted in an event generation module) is sent to the rule verification engine ⑥ (as one of the information sources described before).

3.2. *Process Description at Prosumer Level*

Using YAWL language, managers may define processes in a very easy manner with graphic instruments (see Fig. 2). At the highest level (prosumer level) managers (prosumers) are defining a complex production process or workflow W. This workflow is an ordered sequence T of M_T tasks, connected through M_E oriented edges, E (1). Each task $t_i \in$ is labelled with a discrete timestamp n_i indicating the planned temporal order for the workflow. On the other hand, each edge $e_i \in E$ is labelled with a collection SP_i indicating the list of subproducts flowing (as input and/or outputs) between tasks along the edge (2). Labelling applications λ_T and λ_E are employed to relate labels and edges and tasks. To learn which tasks t_i are connected through oriented edges e_i, and incidence application

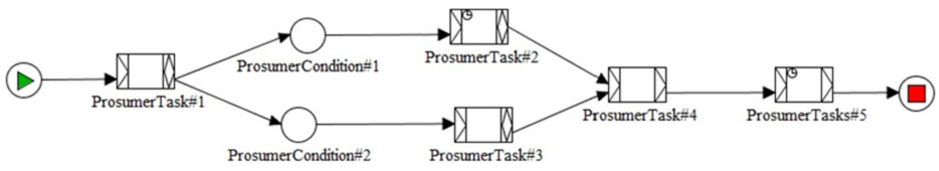

Fig. 2. Graphic representation of a YAWL process.

γ_W is defined (3). This application relates each edge e_i with the order pair of tasks t_i, t_j this edge connects. The edge is outgoing from t_i and incoming in t_j:

$$W = \{T, E\}, \qquad T = \{t_i,\ i = 1, \dots, M_T\}, \qquad E = \{e_i,\ i = 1, \dots, M_E\}, \tag{1}$$

$$\lambda_T(t_i) = n_i, \qquad \lambda_E(e_i) = SP_i = \left\{sp_i^j,\ j = 1, \dots, M_S P\right\}, \tag{2}$$

$$\gamma_W : E \to T \times T, \qquad \gamma_W(e_i) = \{t_i, t_j\}. \tag{3}$$

Besides, in YAWL, tasks may be labelled with a list $\Im_i$ of M_I quality indicators $\amalg_i^j$, employed to accept or reject the actual tasks execution (4). These indicators may refer maximum delays, jitter, errors in physical dimensions, etc. A labelling application λ_{ind} is employed to relate indicators and tasks (5). This list must be exhaustive: every possible indicator must be present in every task. If an indicator does not apply to a particular task, this value in the list should be empty but still present.

$$\Im_i = \left\{\amalg_i^j,\ j = 1, \dots, M_I\right\}, \tag{4}$$

$$\lambda_{ind}\ (t_i) = \Im_i, \tag{5}$$

$$\mathcal{C}_i = \left\{c_i^j,\ j = 1, \dots, M_C\right\}, \qquad \lambda_{cond}\ (e_i)\ = \mathcal{C}_i. \tag{6}$$

Moreover, bifurcation conditions can be considered in YAWL process definitions (see Fig. 2). These conditions are modelled as lists $\mathcal{C}_i$ of semantic annotations associated to edges through a label application λ_{cond} (6). These annotations c_i^j are logical predicates referred to output characteristics, time, etc. This whole process description (including all labels), then, is analysed and decomposed in a specific engine. The first step is to calculate the key process branches or subprocesses ω_i (7):

$$W = \bigcup_i \omega_i. \tag{7}$$

A key subprocess is characterized by only one input point (only one task receives inputs from the physical world) and only one output point (only one task generates a physical service or product). Besides, relations among tasks within the subprocess do not have associated subproducts $SP_i = \emptyset$, but input and output edges to the subprocess do have associated subproducts. Inside a subproduct, then, the temporal order is artificially induced by YAWL language, as it requires all tasks to be connected in sequences. But the process does not require tasks to be executed in that order and it could be altered. When a key subprocess is detected, it is extracted to an independent report to be analysed separately using deformation metrics, and in the global description the subprocess is replaced by a state s_j, taken from a set Σ_W of possible states for this workflow. And a transaction relation ρ_W is built to connect all states s_j according to restrictions in the original process (8). Algorithm 1 describes the proposed mechanism to calculate subprocess and new states and the transaction relation.

On the other hand, annotations $\mathcal{C}_i$ must be transformed into logic predicates referred to states s_j and/or the discrete timestamps n_i. To do that, a semantic algorithm may be employed. This algorithm is not described in this paper, but similar solutions may be found in the state of the art [8]. To do that, a repository of generic logic predicates is considered. These generic predicates are instanced according to annotations $\mathcal{C}_i$ to generate a set $\mathcal{R}$ of atomic propositions r_i or logical rules. An interpretation function L_W is finally defined, indicating what predicates from $\mathcal{R}$ are true for each state s_j (8).

The set of states Σ_W and both the interpretation L_W and the transaction ρ_W applications define a Kripke structure K (9) which is transferred to the business layer for further processing. This Kripke structure is formally defined if only one additional element is considered: the set of states from which the workflow may be initiated Σ_{init}.

$$\rho_W(s_i) = s_{i+1}, \qquad L_W : \Sigma_W \to \mathcal{R}, \qquad L_W(s_j) = \{r_i\}, \tag{8}$$

$$K : \langle \Sigma_W, \Sigma_{init}, \rho_W, L_W \rangle. \tag{9}$$

A second element is generated in the analysis and decomposition engine. For each key subprocess ω_i, each task t_i is labelled with an exhaustive set of indicators $\mathfrak{I}_i$. These indicators are split into two different vectors $\mathfrak{I}_i^a$ and $\mathfrak{I}_i^r$ (10). On the one hand, absolute indicators $\mathfrak{I}_i^a$ referring state variables (such as time, temperature, etc.) are preserved (including empty positions). On the other hand, relative indicators $\mathfrak{I}_i^r$ referring tolerances, errors, etc. are processed to calculate three basic and global deformation metrics: stiffness, strength and ductility.

$$\mathfrak{I}_i = \mathfrak{I}_i^a \cup \mathfrak{I}_i^r, \qquad \begin{aligned} \mathfrak{I}_i^a &= \{\amalg_{i-a}^j j = 1, \dots, M_{I-A}\}, \\ \mathfrak{I}_i^r &= \{\amalg_{i-r}^j j = 1, \dots, M_{I-R}\}. \end{aligned} \tag{10}$$

- Stiffness (F) represents the ability of a process to resist to deformations. In other words, it presents how much a process may absorb variations in the input parameters and conditions and still keep the planned values in the state variables.
- Strength (G) represents the ability of a process to prevent unsatisfactory executions. It describes how resistant the internal organization of the process is, producing correct executions even if inputs and/or state variables suffer great changes with respect to the original model.
- Ductility (D) represents the ability of a process to be deformed and, still, produce correct executions. This parameter indicates how flexible a process is, so even large changes in the state variables are considered as valid executions.

Algorithm 1: Key subprocess identification and Kripke structure creation

```
input  : Workflow $W$ including all labels
output : Key subprocesses $\{\omega_i\}$ and Kripke structure $K$
foreach $n_i \in [1, \infty]$ do
    Obtain $t_i \in W$ such that $\lambda_T(t_i) = n_i$
    foreach $\omega \in \{\omega_i\}$ do
        foreach $t_j \in \omega$ do
            Obtain $e_i n = \gamma_W^{-1}(\{t_i, t_j\})$
            if $\lambda_E(e_{in}) = \emptyset$ and $\gamma_W^{-1}(\{t_i, null\}) = \emptyset$ then
                Add $t_i$ and $e_{in}$ to $\omega$
            else
                Create a new subprocess $\omega_{new}$
                Add $t_i$ and $e_{in}$ to $\omega_{new}$
                Add $\omega_{new}$ to $\{\omega_i\}$
            end
        end
        Obtain $\{e_{out}\} = \gamma_W^{-1}(\{t_j, t_i\}), \forall t_j \in W$
        if $\lambda_E(e_{out}) \neq \emptyset$ or $\gamma_W^{-1}(\{null, t_i\}) \neq \emptyset$ then
            Add $e_{out}$ to $\omega$
            Close subprocess $\omega$
        end
    end
end
Merge subprocesses $\{\omega_i\}$ with common tasks
Instantiate a new Kripke structure $K : \langle \Sigma_W, \Sigma_{init}, \rho_W, L_W \rangle$
foreach $\omega_j \in \{\omega_i\}$ do
    Generate a new state $s_j$ for $\omega_j$
    Add $s_j$ to $\Sigma_W$
    Create the list of logical predicates $\{r_j\}$ from the conditions $\mathcal{C}_j$
    Add $\{r_j\}$ to $\mathcal{R}$
    Add a new relation $L_W(s_j) = \{r_j\}$
    foreach $\omega_k \in \{\omega_i\}$ do
        if $\exists e$ such that $\gamma_W(e) = \{t_k, t_j\}$ being $t_k \in \omega_k$ and $t_j \in \omega_j$ then
            Add a connection between $s_k$ and $s_j$ to $\rho_W$
        end
        if $\exists e$ such that $\gamma_W(e) = \{t_j, t_k\}$ being $t_k = null$ then
            Add $s_j$ to $\Sigma_{init}$
        end
    end
end
```

Stiffness is related to security margins and safeguards between consecutive tasks in the process model (11). As these security margins are larger, it is easier for the process to resist to global deformations, as the effect of inputs on the first tasks are absorbed by security margins. Function calculating the stiffness of a process σ_f must, then, be strictly increasing: linear, logarithmic or exponential functions could be employed depending on the scenario.

Strength is related to the number of restrictions and indicators in the list $\mathfrak{I}_i^r$ (12). As more restrictions are considered, the process is more probable to generate an unsatisfactory execution. Besides, if we consider all restrictions as independent effects, the probability of unsatisfactory executions grows up exponentially with the number of restrictions. Then, function calculating the process strength σ_f is exponentially decreasing.

Finally, ductility is obtained from the tolerances and relative errors in the list $\mathfrak{I}_i^r$ (13). As tolerances go up, processes can be deformed at a higher level, but the process execution is still valid. Then, the function calculating the ductility of the process σ_d is strictly increasing: linear, logarithmic or exponential functions could be employed depending on the scenario

$$F_i = \sigma_f\left(\left|\amalg_{i-a}^{j+1} - \amalg_{i-a}^{j}\right| j = 1, \dots, M_{I-A}\right), \tag{11}$$

$$G_i = \sigma_g(M_{I-R}), \tag{12}$$

$$D_i = \sigma_d\left(\left|\amalg_{i-r}^{j}\right| j = 1, \dots, M_{I-R}\right). \tag{13}$$

Then, the subprocess descriptions $\{\omega_i\}$ together with the vector $\{\mathcal{I}_i^a\}$ and the deformation indicators $\{(F_i, G_i, D_i)_i\}$ are transferred to the production layer for further processing.

3.3. *Global Process Control and Rule Validation at Business Level*

At business level, a Kripke structure (9) describing the process model is received. On the other hand, from this level, the control solution is viewed as a discrete event system (DES), $\mathfrak{Z}$ (14). Where X is the set of possible states in the system; $\mathcal{E}$ is the finite set of events; f is the transition function indicating the next state for a given current state and a received event (15); Γ is the function indicating what events are active for a given state (16) and $\mathcal{P}(\mathcal{E})$ is the power set of $\mathcal{E}$; X_0 is the initial state of the system and X_m is the set of market states (of allowed final states).

$$\mathfrak{Z} = (X, \mathcal{E}, f, \Gamma, X_0, X_m), \tag{14}$$

$$f : X \times \mathcal{E} \rightarrow X, \tag{15}$$

$$\Gamma : X \rightarrow \mathcal{P}(\mathcal{E}). \tag{16}$$

In this DES, $\mathcal{E}^*$ is the Kleene closure of $\mathcal{E}$ which is the set of all possible string in the system (including the empty string); being a string the juxtaposition of states in X describing the evolution of the system; i.e. $\mathcal{E}^*$ is the set of all possible evolutions the system may follow. Two different languages (sets of strings) can be defined, then. The language

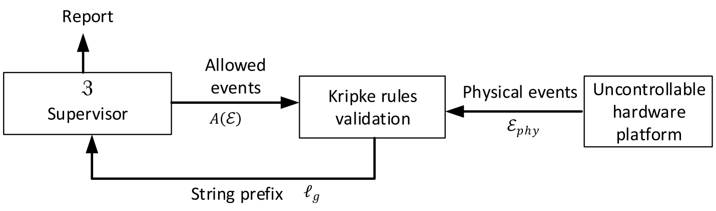

Fig. 3. Supervisory control solution.

marked (17) $\mathcal{L}_m$ contains all strings that finished with a marked state. The language generated (18) $\mathcal{L}_g$ contains all possible strings, i.e. all possible system evolution although they may be not acceptable. In traditional supervisory control systems, the underlaying event-driven infrastructure is assumed to be fully controllable (all events are controllable) or partially controllable (where only some events are uncontrollable). However, in future Industry 4.0 all events tend to be uncontrollable (as devices and people are totally dynamic and autonomous). In that case, common supervisory control theory is not applicable, as its objective is to avoid certain undesired controllable events (and, in Industry 4.0, they are all incontrollable). On the contrary, the objective of the proposed supervisory mechanism is to create reports when undesired states are reached.

$$\mathcal{L}_m(\mathfrak{Z}) = \{\ell_m\} = \left\{\varepsilon^* \in \mathcal{E}^* \middle| \text{ last element in } \varepsilon \in X_m\right\}, \tag{17}$$

$$\mathcal{L}_g(\mathfrak{Z}) = \{\ell_g\} = \left\{\varepsilon^* \in \mathcal{E}^*\right\}. \tag{18}$$

Moreover, in this scenario, events and states in the low-level platform are unknown and, probably, infinite. Then, supervisory control cannot be performed directly over the workers and hardware devices, but over the Kripke structure. Actually, this structure is a fair Kripke structure are two additional conditions are met in every process model:

- Justice or weak fairness requirements: Many states s_j activate the same logical predicate.
- Compassion or strong fairness requirements: If many states s_j have the same logical predicate; many states s_k do not activate that logical predicate.

In that way, a two-level verification is performed in the "rule verification engine" (see Fig. 3). In this solution, states $\mathcal{E}$ in DES $\mathfrak{Z}$ and in the Kripke structure K are the same, although the state set Σ_W in the Kripke structure includes a special state s_{null} to represent unknown states (19). This state s_{null} does not activate any logical predicate (20). The initial states are the same (21). Discrete events ε in the DES $\mathfrak{Z}$ are words ℓ_g of the language general $\mathcal{L}_m(\mathfrak{Z})$ (where the empty string corresponds to the unknown state in the Kripke structure), describing the states the Kripke structure has reached (22) before the current state $s_{current}$. Then, the transition function in the DES may be easily constructed from the transaction relation in K (23):

$$\mathcal{E} = \Sigma_W \cup s_{null}, \tag{19}$$

$$L_W(s_{null}) = \emptyset, \tag{20}$$

$$\Sigma_{init} = X_0, \tag{21}$$

$$\varepsilon_i = \ell_g^i = \{ s_i \mid s_i \in \Sigma_W \wedge s_i \in \rho_W^{-1} \circ \cdots \circ \rho_W^{-1}(s_{current}) \}, \tag{22}$$

$$f(s_i, \epsilon_i) = \rho_W(s_i) = s_{i+1}. \tag{23}$$

In those conditions, the proposed supervisory control algorithm operates as follows. First, physical events $\mathcal{E}_{phy}$ (obtained through the event generation module from physical information and reports) are employed to determine in the Kripke structure which states s_j are allowed in the production process (several of them, because of the justice and compassion requirements). This action is performed using the interpretation function (24). If only one state is allowed, a new DES event ε_i is generated, juxtaposing the new event to the previous string prefix ℓ_g^i (25). This information is introduced as strings' prefix in the DES $\mathfrak{Z}$, and the supervisory control module determines which future states $A(\mathcal{E})$ are allowed by the business model (26) and, eventually, reports if the executed process is incompatible with business requirements (27). Algorithm 2 describes the proposed control solution.

$$\{s_j\} = L_W^{-1}(\{r_j\}), \tag{24}$$

$$\varepsilon_{i+1} = \varepsilon_i | s_{current} = \ell_g^i | s_{current} = \ell_g^{i+1}, \tag{25}$$

$$A(\mathcal{E}) = \{ \varepsilon_i \in \mathcal{E} \text{ such that } \varepsilon_{i+1} | \varepsilon \text{ is prefix of } \ell_m \in \mathcal{L}_m(\mathfrak{Z}) \}, \tag{26}$$

$$A(\mathcal{E}) = \emptyset. \tag{27}$$

Different solutions to validate logic rules have been reported (Leucker, 2017), any of them may be integrated into the proposed mechanism.

3.4. *Subprocess Control Using Deformation Functions at Production Level*

Finally, at production level, a collection of labelled key subprocess $\{\omega_i\}$ and deformation metrics $\{(F_i, G_i, D_i)_i\}$ are received. At this point, every task t_i in any subprocess ω_i is totally defined by the set of absolute quality indicators $\mathfrak{I}_i^a$. This set is a M_{I-A}-dimensional vector which, eventually, can be represented as a point in a generalized M_{I-A}-dimensional phase space. To do that, it is enough to consider the position vector $\overrightarrow{x_i}$ of that point (28) in a general Euclidian space. Besides, in the most common case, every subprocess ω_i is composed of a very large set of tasks, so if the same procedure is repeated for every task, finally the whole subprocess is represented as a point cloud in the phase space. This point cloud may be understood as a discrete solid Ω_i in the phase space (29). This solid, moreover, is flexible, as from the beginning we have assumed that tasks inside each subprocess may be altered (see Fig. 4). In this context, M_{ω_i} is the number of tasks in the subprocess ω_i. In those conditions, the subprocess execution may be understood as a function

Algorithm 2: Supervisory control

```
input  : New physical event $\varepsilon_{phy}$ and current state $s_{current}$
output : Report about process execution evolution
1  Validate logic rules in $\mathcal{R}$ considering $\varepsilon_{phy}$ and obtain the list of valid rules $\{r_j\}$
2  Obtain the list of possible states $\{s_j\} = L_W^{-1}(\{r_j\})$
3  if $\{s_j\} \cap A(\mathcal{E}) \neq \emptyset$ then
4      if $card(\{s_j\}) = 1$ then
5          $s_{current} = s_j$
6          Generate a new event $\varepsilon_{new} = \varepsilon_i | s_{current}$
7          foreach element $\ell_m$ in $\mathcal{L}_m(\mathfrak{Z})$ do
8              if $\varepsilon_{new}$ is prefix of $\ell_m$ then
9                  Add $\ell_m$ to $A(\mathcal{E})$
10             end
11         end
12         if $A(\mathcal{E}) = \emptyset$ then
13             Send an alert for unsatisfactory execution
14         end
15         Send $A(\mathcal{E})$ to the Kripke evaluation module
16     end
17 else
18     Send an alert for unsatisfactory execution
19 end
```

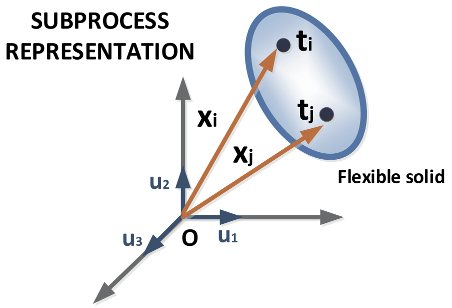

Fig. 4. Subprocess representation in a general phase space.

T_D (30) applied over the solid Ω_i to be transformed into a new and deformed solid Ω_i^* in the original phase space (see Fig. 5). Rotations, extrusions, and any other deformation could appear.

The question to address, then, is if the suffered deformation is high enough to consider whether the execution is unsatisfactory or not.

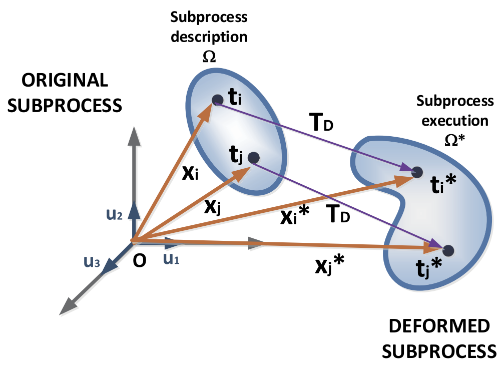

Fig. 5. Subprocess representation in a general phase space.

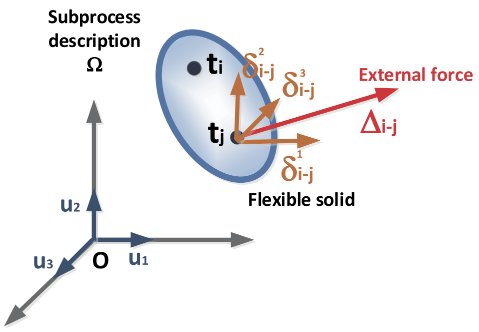

Fig. 6. External forces acting on the flexible solid in the phase space.

$$\overrightarrow{x_i} = \overrightarrow{\Im_i^a O} = \left(\amalg_{i-a}^j,\ j = 1, \dots, M_{I-A}\right), \tag{28}$$

$$\Omega_i = \left\{\overrightarrow{\Im_i^a O},\ i = 1, \dots, M_{\omega_i}\right\}, \tag{29}$$

$$T_D : \Omega_i \subseteq \mathbb{R}^{M_{I-A}} \to \Omega_i^* \subseteq \mathbb{R}^{M_{I-A}}. \tag{30}$$

In the proposed model, a set of external forces Δ_{i-k} (31) is actuating on the solid Ω_i in the point t_k. These forces represent the dynamic, variable, and anarchic behaviour of autonomous devices and humans, which tend to deform the original process model (see Fig. 6). If forces were acting only in one dimension δ_{i-k}^j, the global deformation suffered in that dimension Φ_{i-k}^j could be easily calculated through the Hook law (32); where F_i is the stiffness parameter calculated in the prosumer layer. Although it is not completely correct in terms of physical meaning, we are generalizing this law by taking modular functions, so the global aggregated deformation in all directions of the phase space may be also estimated through the Hook law (33) and, even, the global deformation for the entire solid, by integrating along the entire solid's surface (34). Considering, now, ductility D_i

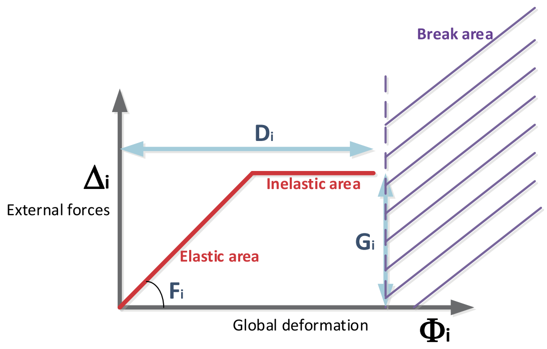

Fig. 7. Generalized Hook law for process verification.

and strength G_i we can extend the common Hook law to include inelastic areas and break zones (where executions are considered unsatisfactory). This new deformation function includes three different branches (35), see Fig. 7.

$$\Delta_{i-k} = \left(\delta_{i-k}^{j},\ j = 1, \ldots, M_{I-A}\right), \tag{31}$$

$$\delta_{i-k}^{j} = F_i \cdot \phi_{i-k}^{j}, \tag{32}$$

$$\|\Delta_{i-k}\| \approx F_i \cdot \|\Phi_{i-k}\|, \tag{33}$$

$$\begin{aligned}\Delta_i &= \sum_{\Omega_i} \|\Delta_{i-k}\| = \frac{1}{M_{\omega_i}} \sum_{k=1}^{M_{\omega_i}} \|\Delta_{i-k}\| \\ &= F_i \cdot \sum_{\Omega_i} \|\Phi_{i-k}\| = F_i \cdot \sum_{k=1}^{M_{\omega_i}} \|\Phi_{i-k}\| = F_i \cdot \Phi_i.\end{aligned} \tag{34}$$

$$\Delta_i = \begin{cases} F_i \cdot \Phi_i, & \text{if } \Phi_i \leqslant \frac{G_i}{F_i}, \\ G_i, & \text{if } \frac{G_i}{F_i} < \Phi_i \leqslant D_i, \\ \text{break} & \text{otherwise.} \end{cases} \tag{35}$$

As can be seen in Fig. 7, it is not necessary to evaluate the external forces Δ_i to learn if global subprocess deformation is too high to consider the execution is valid. These forces are extremely complicated to analyse, as almost no data about them is available. On the contrary, global deformation Φ_i can be evaluated through the absolute quality indicators $\mathfrak{I}_i^a$, which are monitored through the recognition technologies at low-level. For every deformation value Φ_i in the break area, the execution is considered unsatisfactory. Otherwise, it is considered valid. For deformation values in the elastic area, external forces are only deforming the process in a transitory manner, i.e. only some tasks are affected (those which are directly affected by forces). Security margins are absorbing the secondary effects over other tasks, and no permanent global deformation is suffered. On the contrary, when deformation values are in the inelastic area, security margins are not enough to compensate the external forces, and permanent and global deformation appear. Nevertheless, executions are still valid although deformed. Moreover, two contradictory situations must be considered before deformation calculation.

On the one hand, in the inelastic area, small variations in the estimated global deformation Φ_i may cause the subprocess execution to be rejected. So, high precision algorithm to perform those analyses are needed. On the other hand, Industry 4.0 scenarios present real-time requirements, so algorithms must be fast and efficient, and complex high-precision mechanisms must be employed.

Thus, we proposed a two phases deformation evaluation: in the first step, a very simple algorithm where all state variables and quality indicators are considered independent is employed; only if this algorithm places the global deformation in the inelastic area, a second step where a complex and much more precise algorithm is employed would be triggered.

To calculate the first and approximate global deformation Φ_i^{ind} of the subprocess ω_i after an execution, we are considering all quality indicators $\Im_i^a$ to be totally independent. Then, deformation ϕ_{i-k}^j along each indicator and on each task can be calculated independently. This calculation is based on the ratio between the planned value for each quality indicator $\amalg_{i-a}^j$, and the final obtained result $\amalg_{i-a}^{j,*}$ (36). Using this radio, deformation ϕ_{i-k}^j can be obtained through different expressions: unitary deformation (37), Almansi deformation (38), Green deformation (39), Hencky deformation (40) or any other application specific algorithm. Each one of these expressions is adequate for a range of deformation values (see Table 2). Then, we are always initially calculating the unitary deformation, and, after that, we can recalculate the value using a more adequate expression if needed.

$$\mu_{i-k}^j = \frac{\amalg_{i-a}^j}{\amalg_{i-a}^{j,*}}, \tag{36}$$

$$\phi_{i-k}^j = \mu_{i-k}^j - 1, \tag{37}$$

$$\phi_{i-k}^j = \frac{1}{2}\left(1 - \frac{1}{(\mu_{i-k}^j)^2}\right), \tag{38}$$

$$\phi_{i-k}^j = \frac{1}{2}\left((\mu_{i-k}^j)^2 - 1\right), \tag{39}$$

$$\phi_{i-k}^j = \ln(\mu_{i-k}^j). \tag{40}$$

As can be seen in Table 2, using the unitary deformation we can evaluate the subprocess rotation and absolute deformation. Then, for processes where rotation and large deformation values are present, Hencky function is the most precise calculation method. Equally, for null rotation and large deformation values, specific logarithmic algorithms and expressions may be proposed depending on the scenario. Finally, for small values of deformation we can use the Green or Almansi function (as desired) if rotation is present, or the unitary deformation if no rotation and small deformation values are observed.

After calculating the global deformation Φ_i for the subprocess, if this value is in the elastic area, the execution is accepted. If the subprocess execution is in the break area the execution is rejected; and if the value is in the inelastic area, a more precise evaluation is needed. In this more precise calculation mechanism, we are assuming the realistic scenario, where quality indicators are dependent on each other. Then, deformation parameters are obtained through the deformation tensor Ψ_{i-k} (41), where differential expressions are approximated using numerical expressions (42). Basically, for initial task we are employing the previous expressions (37)–(40) to start the calculation algorithm. Other tasks are evaluated through the backward differences. There exists also a tensor expression for Almansi deformation (43) and for Green–Lagrange deformation (44) which can be employed for depending on the scenario under study (selecting that which is closer to reality in all cases).

Module calculation (34), in this case, will refer to the tensor module calculation; contrary to the previous case where vector module was employed.

$$\Phi_{i-k} = \Psi_{i-k} = \nabla T_D = \begin{pmatrix} \frac{\partial \text{Ш}_{i-a}^{1,*}}{\partial \text{Ш}_{i-a}^{1}} & \cdots & \frac{\partial \text{Ш}_{i-a}^{1,*}}{\partial \text{Ш}_{i-a}^{M_{I-A}}} \\ & \cdots & \\ \frac{\partial \text{Ш}_{i-a}^{M_{I-A},*}}{\partial \text{Ш}_{i-a}^{1}} & \cdots & \frac{\partial \text{Ш}_{i-a}^{M_{I-A},*}}{\partial \text{Ш}_{i-a}^{M_{I-A}}} \end{pmatrix}, \tag{41}$$

$$\frac{\partial \text{Ш}_{i-a}^{k,*}}{\partial \text{Ш}_{i-a}^{r}} = \frac{\text{Ш}_{i-a}^{k,*} - \text{Ш}_{i-a}^{k-1,*}}{\text{Ш}_{i-a}^{r} - \text{Ш}_{i-a}^{r-1}}, \quad \text{for } r, k > 1, \tag{42}$$

$$\Phi_{i-k} = \frac{1}{2}\left(1 - \Psi_{i-k}^{-T}\Psi_{i-k}^{-1}\right), \tag{43}$$

$$\Phi_{i-k} = \frac{1}{2}\left(\Psi_{i-k}^{T}\Psi_{i-k} - 1\right). \tag{44}$$

With this much more precise value, the caused deformation is finally evaluated. If it remains in the inelastic area, the execution is approved. In the contrary event, the execution is rejected. Algorithm 3 describes the entire described solution at production level.

4. Experimental Validation

In order to evaluate the performance of the proposed solution, an experimental validation was designed and carried out. The experiments were based on pervasive sensing and computing platforms, deployed in a laboratory, emulating an Industry 4.0 manufacturing environment. In this environment, workers and work automation devices were expected to perform a certain workflow along the day, composed of certain production activities, taken from food manufacturing companies such as baking companies (see Table 4). These production activities, besides, are composed of a quite large collection of tasks, which are finally monitored.

```
Algorithm 3: Deformation evaluation
input  : List of planned quality indicators $\Im_i^a$ and list of finally obtained
         indicators $\Im_i^{a,*}$
output : Report about process execution
1  Create a Boolean variable rotation set to true
2  foreach indicator $\amalg_{i-a}^j$ in the list $\Im_i^a$ do
3      Calculate $\mu_{i-k}^j = \frac{\amalg_{i-a}^j}{\amalg_{i-a}^{j,*}}$
4      Calculate $\phi_{i-k}^j = \mu_{i-k}^j - 1$
5      if $|\phi_{i-k}^j - \phi_{i-k+1}^j| > 10^{-4}$ then
6          Set rotation to false
7      end
8  end
9  foreach $\phi_{i-k}^j$ in $\Phi_i$ do
10     if $0.1 < |\phi_{i-k}^j| < 10$ and rotation then
11         Calculate $\phi_{i-k}^j = \frac{1}{2}((\mu_{i-k}^j)^2 - 1)$
12     else
13         if rotation then
14             Calculate $\phi_{i-k}^j = ln(\mu_{i-k}^j)$
15         end
16     end
17 end
18 Calculate $\Phi_i = \sum_{k=1}^{M_{\omega_i}} \|\Phi_{i-k}\|$
19 if $\frac{G_i}{F_i} < \Phi_i \leqslant D_i$ then
20     foreach $k \in [2, M_{I-A}]$ do
21         $\frac{\partial \amalg_{i-a}^{k,*}}{\partial \amalg_{i-a}^{r}} = \frac{\amalg_{i-a}^{k,*} - \amalg_{i-a}^{k-1,*}}{\amalg_{i-a}^{r} - \amalg_{i-a}^{r-1}}$
22     end
23     Calculate matrix $\Psi_{i-k}$
24     Calculate $\Phi_i = \sum_{k=1}^{M_{\omega_i}} \|\Psi_{i-k}\|$
25 end
26 if $\Phi_i \leqslant \frac{G_i}{F_i}$ then
27     Approve the subprocess execution
28 else
29     Reject the subprocess execution
30 end
```

The scenario was deployed at Universidad Politécnica de Madrid, where a 30 m2 space was conditioned as working environment. Because of sanitary restrictions during 2020 in

Table 2
Different deformation calculation expressions.

Rotation	Deformation: Large	Deformation: Small ($0.1 < \lvert\phi^j_{i-k}\rvert < 10$)
Yes ($\lvert\phi^j_{i-k} - \phi^j_{i-k+1}\rvert < 10^{-4}\ \forall k$)	Hencky deformation $\phi^j_{i-k} = \ln(\mu^j_{i-k})$	Green deformation $\phi^j_{i-k} = \frac{1}{2}\left((\mu^j_{i-k})^2 - 1\right)$ Almasi deformation $\phi^j_{i-k} = \frac{1}{2}\left(1 - \frac{1}{(\mu^j_{i-k})^2}\right)$
No	Ad hoc logarithmic definitions	Unitary deformation $\phi^j_{i-k} = \mu^j_{i-k} - 1$

Europe, only three people could be at the same time in the laboratory. Then, if a large amount of people were involved in the experiment, different experiment realization (each one with only three participants) would be performed. Electronic devices present in this installation were configured to simulate a food production environment.

Three different information sources and devices were considered in order to monitor people and device activities: RFID tags, infrared barriers and accelerometers and general sensors (temperature, humidity, light, etc.). Table 3 shows the composition of the deployed infrastructure. All these elements were built together with an Arduino Nano platform and connected through Bluetooth technologies to a gateway supported by a Raspberry Pi device. This gateway, finally, communicates all information to a central server in the cloud for data storage and back-end deployment. The server was deployed in a Linux architecture (Ubuntu 18.04 LTS) with the following hardware characteristics: Dell R540 Rack 2U, 96 GB RAM, two processors Intel Xeon Silver 4114 2.2G, HD 2TB SATA 7,2K rpm.

Autonomous devices in the proposed scenario belonged to three different types (see Table 3):

- Autonomous robots: Cleaning robots and software-based robots, acting in an autonomous manner to help workers in the management of production facility. Software-based robots consisted on agents interacting with virtual PLCs simulating bakery processes, service interruptions, etc.
- Electronic ink displays: Low-cost and low consumption devices employed to display information about next activities to be performed, warning and alerts, etc. All these displays were controlled from the Raspberry local gateway.
- Cyber-Physical Systems and pervasive system to control living conditions: All environmental conditions, such as temperature or humidity, were monitored by pervasive computing platforms and traditional feedback control loops.

In order to recognize activities being performed, two different technologies were employed. To recognize human activities, we are using a two-phase solution composed of different machine learning and pattern recognition layers (Wonham and Cai, 2019). To recognize activities performed by autonomous devices we employed previous works on artificial intelligence mechanisms for Internet-of-Things applications based on signal processing (Bordel *et al.*, 2020b). These components were deployed together with the proposed

Table 3
Infrastructure composition during the experimental validation.

Infrastructure subsystem	Device	Quantity
Monitoring subsystem	RFID tags	19
	Infrared barrier	6
	Accelerometers and general sensors	33
Pervasive platform	Autonomous robots	4
	Electronic ink displays	12
	CPS for the living environment	4

Table 4
Most common production activities in the experimental validation.

Workflow	Most important activities
Supervising bakery products	Visual inspection of products, quality assessment and product discard.
Supervising bakery production	Monitoring of production processes, operating status, controlling speed, stops and notifications.
Pastry packaging	Dispense, grouping, labeling and packaging.
Auxiliary operations in food industry	Customer service, warehouse cleaning, etc.

solution in the referred cloud server. This server also offered a prosumer webpage, where YAWL-based workflows of production activities could be generated. Workflows are based in bakery production activities of the state of the art (Katz *et al.*, 1963), and most common activities are showed in Table 4.

The experiment considered four different workflows with a variable number of tasks and production activities. All workflows were executed by autonomous devices and twelve people in an eight-hour session (standard labour schedule). People were selected as a homogenous community, with respect to the gender and age parity. The experiment included three different phases:

- Training phase: Participants received some training about the activities and workflows they had to perform and the context and experiment conditions.
- Process execution phase: Each participant was interviewed in this phase. The participant was asked to execute a certain process, which is described in a short document, as any company would do.
- Evaluation phase: In this phase, experts evaluated the records obtained by the system about the validity of executions, and they were compared to observations made by experts (which finally determined whether an execution was correct or not).

Two different experiments were performed using the described infrastructure. During the first experiment, the precision and success rate of the proposed supervisory control mechanism was evaluated. The number (percentage) of activities and workflows correctly detected as successful or unsatisfactory executions is calculated, including the percentage of false positives and false negatives. Experiments are repeated for workflows with different number of tasks, and they are also compared to traditional top-down control approaches (Bordel *et al.*, 2018c). During the second experiment, the scalability of the proposed control solution with respect to the complexity (number of tasks) of the executed

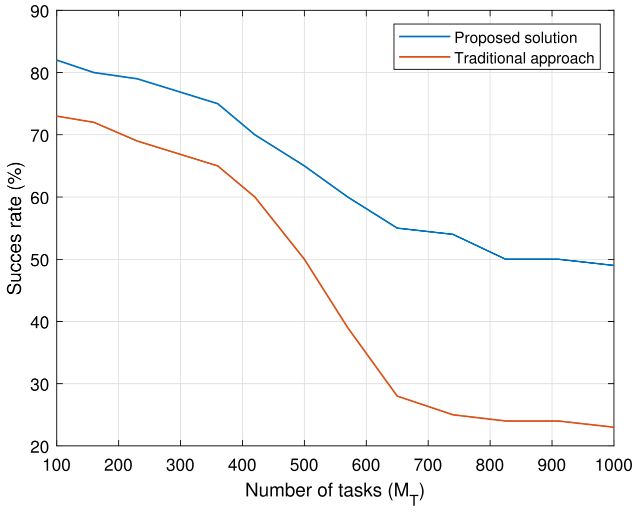

Fig. 8. Execution correctly valuated and accepted (or rejected).

workflow and quality indicators in the task description is analysed. The main indicator analysed in this second experiment is the processing delay.

5. Results and Discussion

Figure 8 shows the results from the first experiment, comparing the success rate in the proposed control mechanism (rule verification engine) ⑥ to previously existing traditional solutions [5], for workflows with a different number of tasks $\mathbf{M_T}$.

As can be seen, the proposed supervisory control mechanism reaches a higher successful rate in all cases. In a first zone, for workflows where $\mathbf{M_T} < \mathbf{400}$, the success rate is improved round 10% with respect to traditional control solutions, reaching a success rate of 82% (approximately). However, as the number of activities in the workflow goes up, and the success rate goes down for both approaches. In fact, a higher number of activities implies a higher variability and makes it much more complex to control anarchic executions. In any case, traditional solutions get worse faster than the innovative proposed mechanism. Thus, for workflows with a number of tasks $\mathbf{M_T} = \mathbf{600}$, the proposed solution presents a success rate of 58% (approximately), while traditional mechanisms present a performance of almost a 300% worse. After this point, both mechanisms reach the minimum precision and they remain stable.

An interesting result to be evaluated is how the wrongly evaluated process executions are split into false positive detection and false negative detections. Figure 9 shows those results.

In general, as can be seen, traditional approaches present a much higher number of false negative detections (executions that are valid are considered unsatisfactory), while

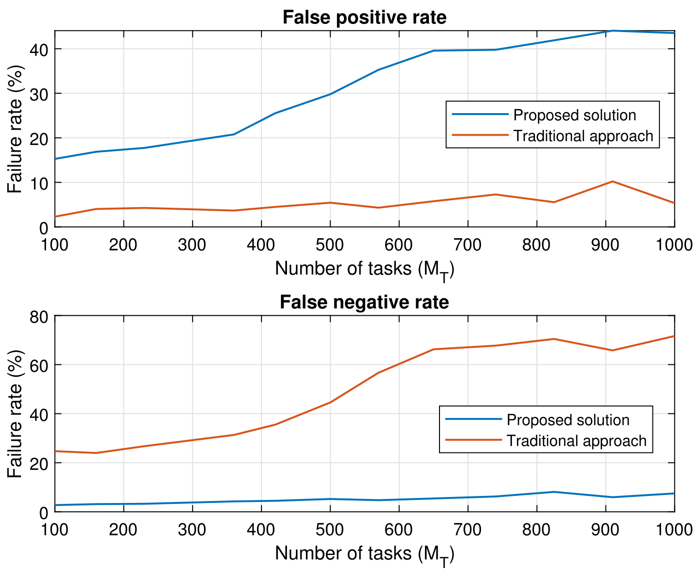

Fig. 9. Distribution of errors in the first experiment.

the proposed solution tends to create false positive evaluations (executions that are not valid but are finally accepted) in a higher rate than state of the art technologies.

Finally, Fig. 10 shows the results from the second experiment. The scalability in terms of processing delay (from deformation calculation ⑫ to rule verification ⑥) is analysed, considering different number of tasks M_T and a number of absolute state variables and quality indicators M_{I-A}. As can be seen, evolution of processing delay with respect to the number of tasks in the workflow is almost linear. And, then, the temporal order of the proposed algorithm in that variable is $\mathcal{O}(n)$. On the other hand, processing delay according to the number of quality indicators describing each task M_{I-A} follows a function with a higher growing rate than linear functions, which can be approximated by $\mathcal{O}(n \cdot \log(n))$. In both cases we are avoiding exponential or high-order polynomial temporal order which could make the proposed solution impractical for high complexity Industry 4.0 production scenarios. Processing delay, in any case, is in the order of hundreds of milliseconds.

6. Conclusions

Industry 4.0 solutions are composed of autonomous engineered systems where heterogenous agents act in a choreographed manner to create complex workflows. In this work, supervisory control techniques are employed to guarantee a correct execution of distributed and choreographed processes in Industry 4.0 scenarios. At prosumer level, processes are represented using soft models where logic rules and deformation indicators are used to analyse the correctness of executions. These logic rules are verified using specific engines at business level. These engines are fed with deformation metrics obtained through tensor

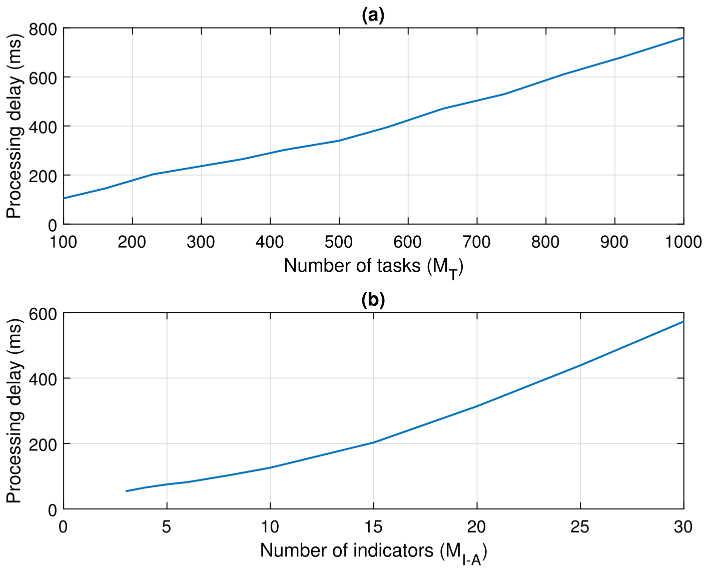

Fig. 10. Processing delay evolution: scalability. (a) Number of tasks in the workflow. (b) Number of quality indicators describing each task.

deformation functions at production level. To apply deformation functions, processes are represented as discrete flexible solids in a phase space, under external forces representing the variations in every task's inputs.

Experimental validation shows that the proposed mechanism improves the performance of traditional control solutions in a percentage between 10% and 300%, depending on the workflow to be executed. Besides, scalability of the proposed solution is almost linear with respect to the number of tasks in the workflow, and $n \cdot \log(n)$ with respect to the number of quality indicators or state variables describing tasks.

As future works, it is intended to do a proof of concept in a relevant environment, a real food processing facility for the monitoring of production activities in a non-intrusive way. For this, it is necessary to consider, in addition to the technical challenges of integration with sensorization devices present in this relevant environment, ethical and privacy considerations for the workers who are in these facilities. For these tests and the continuation of this line of research we have a research framework with relevant food processing companies, established in accordance with the DEMETER project.

Funding

The research leading to these results has received funding from DEMETER project (H2020-DT-2018-2020. Grant no: 857202).

References

Aminifar, A. (2016). *Analysis, Design, and Optimization of Embedded Control Systems*, Vol. 1746. Linköping University Electronic Press, Linköping, Sweden. https://doi.org/10.3384/diss.diva-124319.

Aven, T. (2017). How some types of risk assessments can support resilience analysis and management. *Reliability Engineering and System Safety*, 167, 536–543. https://doi.org/10.1016/j.ress.2017.07.005.

Bagheri, B., Yang, S., Kao, H.A., Lee, J. (2015). Cyber-physical systems architecture for self-aware machines in Industry 4.0 environment. In: *IFAC-PapersOnLine*, Vol. 28. Elsevier, Berlin, pp. 1622–1627. https://doi.org/10.1016/j.ifacol.2015.06.318.

Bensoussan, A., Çakanyildirim, M., Sethi, S.P. (2009). Optimal ordering policies for inventory problems with dynamic information delays. *Production and Operations Management*, 16(2), 241–256. https://doi.org/10.1111/j.1937-5956.2007.tb00178.x.

Bordel, B., Alcarria, R. (2017). Assessment of human motivation through analysis of physiological and emotional signals in Industry 4.0 scenarios. *Journal of Ambient Intelligence and Humanized Computing*, 1, 1–21. https://doi.org/10.1007/s12652-017-0664-4.

Bordel, B., Alcarria, R., Jara, A. (2017a). Process execution in humanized cyber-physical systems: Soft processes. In: *Iberian Conference on Information Systems and Technologies, CISTI*. IEEE Computer Society, Washington DC, USA. https://doi.org/10.23919/CISTI.2017.7975901.

Bordel, B., Alcarria, R., Robles, T., Martín, D. (2017b). Cyber-physical systems: extending pervasive sensing from control theory to the internet of things. *Pervasive and Mobile Computing*, 40, 156–184. https://doi.org/10.1016/j.pmcj.2017.06.011.

Bordel, B., Alcarria, R., Martín, D., Robles, T., de Rivera, D.S. (2017c). Self-configuration in humanized cyber-physical systems. *Journal of Ambient Intelligence and Humanized Computing*, 8(4), 485–496. https://doi.org/10.1007/s12652-016-0410-3.

Bordel, B., Miguel, C., Alcarria, R., Robles, T. (2018a). A hardware-supported algorithm for self-managed and choreographed task execution in sensor networks. *Sensors*, 18(3), 812. https://doi.org/10.3390/s18030812.

Bordel, B., Alcarria, R., Sanchez De Rivera, D., Martín, D., Robles, T. (2018b). Fast self-configuration in service-oriented smart environments for real-time applications. *Journal of Ambient Intelligence and Smart Environments*, 10(2), 143–167. https://doi.org/10.3233/AIS-180479.

Bordel, B., Alcarria, R., de Rivera, D.S., Robles, T. (2018c). Process execution in cyber-physical systems using cloud and cyber-physical internet services. *Journal of Supercomputing*, 74(8), 4127–4169. https://doi.org/10.1007/s11227-018-2416-4.

Bordel, B., Alcarria, R., Sánchez-de-Rivera, D. (2019). A two-phase algorithm for recognizing human activities in the context of industry 4.0 and human-driven processes. In: *Advances in Intelligent Systems and Computing*, Vol. 931. Springer-Verlag, Berlin, pp. 175–185. https://doi.org/10.1007/978-3-030-16184-2_18.

Bordel, B., Alcarria, R., Robles, T. (2020a). Supervising industrial distributed processes through soft models, deformation metrics and temporal logic rules. In: *Advances in Intelligent Systems and Computing*, Vol. 1160 AISC. Springer, Berlin, pp. 125–136. https://doi.org/10.1007/978-3-030-45691-7_12.

Bordel, B., Alcarria, R., Robles, T., You, I. (2020b). A predictor-corrector algorithm based on laurent series for biological signals in the internet of medical things. *IEEE Access*, 8, 109360–109371. https://doi.org/10.1109/ACCESS.2020.3001275.

Boyer, S.A. (2016). *SCADA: Supervisory Control and Data Acquisition, Fourth Edition*. International Society of Automation, Pennsylvania, USA, pp. 257.

Branger, J., Pang, Z. (2015). From automated home to sustainable, healthy and manufacturing home: a new story enabled by the Internet-of-Things and Industry 4.0. *Journal of Management Analytics*, 2(4). https://doi.org/10.1080/23270012.2015.1115379.

Calderón Godoy, A., González Pérez, I. (2018). Integration of sensor and actuator networks and the SCADA system to promote the migration of the legacy flexible manufacturing system towards the Industry 4.0 concept. *Journal of Sensor and Actuator Networks*, 7(2), 23. https://doi.org/10.3390/jsan7020023.

Chhetri, S.R., Rashid, N., Faezi, S., Faruque, M.A.A. (2017). Security trends and advances in manufacturing systems in the era of Inxdustry 4.0. In: *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD*, Vol. 2017-November. Institute of Electrical and Electronics Engineers Inc., New Jersey, USA, pp. 1039–1046. https://doi.org/10.1109/ICCAD.2017.8203896.

Diez-Olivan, A., Del Ser, J., Galar, D., Sierra, B. (2019). Data fusion and machine learning for industrial prognosis: trends and perspectives towards Industry 4.0. *Information Fusion*, 50, 92–111. https://doi.org/10.1016/j.inffus.2018.10.005.

Disney, S.M., Towill, D.R., Warburton, R.D.H. (2006). On the equivalence of control theoretic, differential, and difference equation approaches to modeling supply chains. *International Journal of Production Economics*, 101(1 spec. iss.), 194–208. https://doi.org/10.1016/j.ijpe.2005.05.002.

Dolgui, A., Ivanov, D., Sethi, S.P., Sokolov, B. (2019). Scheduling in production, supply chain and Industry 4.0 systems by optimal control: fundamentals, state-of-the-art and applications. *International Journal of Production Research*, 57(2), 411–432. https://doi.org/10.1080/00207543.2018.1442948.

Ebling, M.R., Want, R. (2017). *Pervasive Computing Revisited*. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/MPRV.2017.2940959.

Fabre, E., Jezequel, L. (2009). Distributed optimal planning: an approach by weighted automata calculus. In: *Proceedings of the IEEE Conference on Decision and Control*. IEEE, New Jersey, USA, pp. 211–216. https://doi.org/10.1109/CDC.2009.5400084.

Foruzan, E., Asgarpoor, S., Bradley, J.M. (2016). Hybrid system modeling and supervisory control of a microgrid. In: *NAPS 2016 – 48th North American Power Symposium, Proceedings*. Institute of Electrical and Electronics Engineers Inc., New Jersey, USA. https://doi.org/10.1109/NAPS.2016.7747840.

Frazzon, E.M., Kück, M., Freitag, M. (2018). Data-driven production control for complex and dynamic manufacturing systems. *CIRP Annals*, 67(1), 515–518. https://doi.org/10.1016/j.cirp.2018.04.033.

Garcia, C.A., Ibeas, A., Herrera, J., Vilanova, R. (2012). Inventory control for the supply chain: an adaptive control approach based on the identification of the lead-time. *Omega*, 40(3), 314–327. https://doi.org/10.1016/j.omega.2011.07.003.

Geiger, M., Harrer, S., Lenhard, J., Wirtz, G. (2018). BPMN 2.0: the state of support and implementation. *Future Generation Computer Systems*, 80, 250–262. https://doi.org/10.1016/j.future.2017.01.006.

Golob, M., Bratina, B. (2018). Web-based control and process automation education and Industry 4.0. *International Journal of Engineering Education*, 34(4), 1199–1211.

Gonzalez, A.G.C., Alves, M.V.S., Viana, G.S., Carvalho, L.K., Basilio, J.C. (2018). Supervisory control-based navigation architecture: a new framework for autonomous robots in Industry 4.0 environments. *IEEE Transactions on Industrial Informatics*, 14(4), 1732–1743. https://doi.org/10.1109/TII.2017.2788079.

Goryca, J., Hill, R.C. (2013). Formal synthesis of supervisory control software for multiple robot systems. In: *Proceedings of the American Control Conference*. Institute of Electrical and Electronics Engineers Inc., New Jersey, USA, pp. 125–131. https://doi.org/10.1109/acc.2013.6579825.

Gu, C., Wang, X., Li, Z. (2019). Synthesis of supervisory control with partial observation on normal state-tree structures. *IEEE Transactions on Automation Science and Engineering*, 16(2), 984–997. https://doi.org/10.1109/TASE.2018.2880178.

Hill, R.C., Lafortune, S. (2017). Scaling the formal synthesis of supervisory control software for multiple robot systems. In: *Proceedings of the American Control Conference*. Institute of Electrical and Electronics Engineers Inc., New Jersey, USA, pp. 3840–3847. https://doi.org/10.23919/ACC.2017.7963543.

Igure, V.M., Laughter, S.A., Williams, R.D. (2006). Security issues in SCADA networks. *Computers and Security*, 25(7), 498–506. https://doi.org/10.1016/j.cose.2006.03.001.

Iqbal, J., Afzal Khan, S., Ahmad Zafar, N., Ahmad, F. (2012). *Modeling Supervisory Control of Autonomous Mobile Robots Using Graph Theory, Automata and Z Notation*. Technical Report 12.

Ivanov, D., Dolgui, A., Sokolov, B., Werner, F., Ivanova, M. (2016a). A dynamic model and an algorithm for short-term supply chain scheduling in the smart factory Industry 4.0. *International Journal of Production Research*, 54(2), 386–402. https://doi.org/10.1080/00207543.2014.999958.

Ivanov, D., Dolgui, A., Sokolov, B., Werner, F. (2016b). Schedule robustness analysis with the help of attainable sets in continuous flow problem under capacity disruptions. *International Journal of Production Research*, 54(11), 3397–3413. https://doi.org/10.1080/00207543.2015.1129467.

Ivanov, D., Dolgui, A., Sokolov, B. (2019). The impact of digital technology and Industry 4.0 on the ripple effect and supply chain risk analytics. *International Journal of Production Research*, 57(3), 829–846. https://doi.org/10.1080/00207543.2018.1488086.

Katz, S., Ford, A.B., Moskowitz, R.W., Jackson, B.A., Jaffe, M.W. (1963). Studies of illness in the aged: the index of ADL: a standardized measure of biological and psychosocial function. *JAMA: The Journal of the American Medical Association*, 185(12), 914–919. https://doi.org/10.1001/jama.1963.03060120024016.

Kloetzer, M., Mahulea, C. (2016). Multi-robot path planning for syntactically co-safe LTL specifications. In: *2016 13th International Workshop on Discrete Event Systems, WODES 2016*. Institute of Electrical and Electronics Engineers Inc., New Jersey, USA, pp. 452–458. https://doi.org/10.1109/WODES.2016.7497887.

Kretschmer, F., Friedl, S., Lechler, A., Verl, A. (2016). Communication extension for cloud-based machine control of simulated robot processes. In: *Proceedings of the IEEE International Conference on Industrial Tech-*

nology, Vol. 2016-May. Institute of Electrical and Electronics Engineers Inc., New Jersey, USA, pp. 54–58. https://doi.org/10.1109/ICIT.2016.7474725.

Lalwani, C.S., Disney, S.M., Towill, D.R. (2006). Controllable, observable and stable state space representations of a generalized order-up-to policy. *International Journal of Production Economics*, 101(1 spec. iss.), 172–184. https://doi.org/10.1016/j.ijpe.2005.05.014.

Lasi, H., Fettke, P., Feld, T., Hoffmann, M. (2014). Industry 4.0. *Business & Information Systems Engineering*, 6(4).

Leucker, M. (2017). Runtime verification for linear-time temporal logic. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 10215. Springer-Verlag, Berlin, pp. 151–194. https://doi.org/10.1007/978-3-319-56841-6_5.

Lin, J., Spiegler, V.L.M., Naim, M.M. (2018). Dynamic analysis and design of a semiconductor supply chain: a control engineering approach. *International Journal of Production Research*, 56(13), 4585–4611. https://doi.org/10.1080/00207543.2017.1396507.

Lu, Y. (2017). Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration*, 6, 1–10. https://doi.org/10.1016/j.jii.2017.04.005.

Meissner, H., Ilsen, R., Aurich, J.C. (2017). Analysis of control architectures in the context of Industry 4.0. *Procedia CIRP*, 62, pp. 165–169. https://doi.org/10.1016/j.procir.2016.06.113.

Merchan, D.F., Peralta, J.A., Vazquez-Rodas, A., Minchala, L.I., Astudillo-Salinas, D. (2018). Open source SCADA system for advanced monitoring of industrial processes. In: *Proceedings – 2017 International Conference on Information Systems and Computer Science, INCISCOS 2017*, Vol. 2017-November. Institute of Electrical and Electronics Engineers Inc., New Jersey, USA, pp. 160–165. https://doi.org/10.1109/INCISCOS.2017.9.

Mohammad, U., Yee Low, C., Abd Rahman, R., Akmal Johar, M., Theng Koh, C., Dumitrescu, R., Rabe, M., Asmar, L., Kamaruddin, S. (2019). Smart factory reference model for training on Industry 4.0. *Journal of Mechanical Engineering*, 16(2), 129–144.

Moreira, M.V., Basilio, J.C. (2014). Bridging the gap between design and implementation of discrete-event controllers. *IEEE Transactions on Automation Science and Engineering*, 11(1), 48–65. https://doi.org/10.1109/TASE.2013.2281733.

Nguyen, H.T., Walker, C.L., Walker, E.A. (2018). *A First Course in Fuzzy Logic*, 4th ed. Chapman and Hall/CRC, United Kingdom, pp. 448.

Pan, Y., Yang, G.H. (2017). Event-triggered fuzzy control for nonlinear networked control systems. *Fuzzy Sets and Systems*, 329, 91–107. https://doi.org/10.1016/j.fss.2017.05.010.

Rossit, D.A., Tohmé, F., Frutos, M. (2019). Industry 4.0: smart scheduling. *International Journal of Production Research*, 57(12), 3802–3813. https://doi.org/10.1080/00207543.2018.1504248.

Roszkowska, E. (2002). Supervisory control for multiple mobile robots in 2D space. In: *Proceedings of the 3rd International Workshop on Robot Motion and Control, RoMoCo 2002*. Institute of Electrical and Electronics Engineers Inc., New Jersey, USA, pp. 187–192. https://doi.org/10.1109/ROMOCO.2002.1177106.

Sajid, A., Abbas, H., Saleem, K. (2016). Cloud-assisted IoT-based SCADA systems security: a review of the state of the art and future challenges. *IEEE Access*, 4, 1375–1384. https://doi.org/10.1109/ACCESS.2016.2549047

Sampath, M., Sinnamohideen, K., Lafortune, S., Teneketzis, D. (1995). Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, 40(9), 1555–1575. https://doi.org/10.1109/9.412626.

Sánchez, B.B., Alcarria, R., Sańchez-De-Rivera, D., Sańchez-Picot, Á. (2016). Enhancing process control in Industry 4.0 scenarios using Cyber-Physical systems. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 7(4), 41–64. https://doi.org/10.22667/JOWUA.2016.12.31.041.

Scholz-Reiter, B., Dashkovskiy, S., Görges, M., Naujok, L. (2011). Stability analysis of autonomously controlled production networks. *International Journal of Production Research*, 49(16), 4857–4877. https://doi.org/10.1080/00207543.2010.505215.

Silva, D.R.C., Oliveira, G.M.B., Silva, I., Ferrari, P., Sisinni, E. (2018). Latency evaluation for MQTT and WebSocket protocols: an Industry 4.0 perspective. In: *Proceedings – IEEE Symposium on Computers and Communications*, Vol. 2018-June. Institute of Electrical and Electronics Engineers Inc., New Jersey, USA, pp. 1233–1238. https://doi.org/10.1109/ISCC.2018.8538692.

Sokolov, B., Dolgui, A., Ivanov, D. (2018). Optimal control algorithms and their analysis for short-term scheduling in manufacturing systems. *Algorithms*, 11(5), 57. https://doi.org/10.3390/a11050057.

Spiegler, V.L.M., Naim, M.M. (2017). Investigating sustained oscillations in nonlinear production and inventory control models. *European Journal of Operational Research*, 261(2), 572–583. https://doi.org/10.1016/j.ejor.2017.02.010.

Theorin, A., Bengtsson, K., Provost, J., Lieder, M., Johnsson, C., Lundholm, T., Lennartson, B. (2017). An event-driven manufacturing information system architecture for Industry 4.0. *International Journal of Production Research*, 55(5), 1297–1311. https://doi.org/10.1080/00207543.2016.1201604.

Van Der Aalst, W.M.P., Ter Hofstede, A.H.M. (2005). YAWL: Yet another workflow language. *Information Systems*, 30(4), 245–275. https://doi.org/10.1016/j.is.2004.02.002.

Wollschlaeger, M., Sauter, T., Jasperneite, J. (2017). The future of industrial communication: automation networks in the era of the internet of things and Industry 4.0. *IEEE Industrial Electronics Magazine*, 11(1), 17–27. https://doi.org/10.1109/MIE.2017.2649104.

Wonham, W.M., Cai, K. (2019). *Supervisory Control of Discrete-Event Systems*. Communications and Control Engineering. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-77452-7.

Wonham, W.M., Cai, K., Rudie, K. (2018). Supervisory control of discrete-event systems: a brief history. *Annual Reviews in Control*, 45, 250–256. https://doi.org/10.1016/j.arcontrol.2018.03.002.

Yang, T., Fan, W. (2016). Information management strategies and supply chain performance under demand disruptions. *International Journal of Production Research*, 54, 8–27. https://doi.org/10.1080/00207543.2014.991456.

Zhang, Q., Liu, P., Pannek, J. (2019). Combining MPC and integer operators for capacity adjustment in job-shop systems with RMTs. *International Journal of Production Research*, 57(8), 2498–2513. https://doi.org/10.1080/00207543.2018.1521022.

B. Bordel received the BS degree in telecommunication engineering, in 2012, and the MS in telecommunication engineering, in 2014, both from Technical University of Madrid. He obtained his PhD, in 2018, and he is currently an assistant professor in the Computer Science School. His research interests include cyber-physical systems, wireless sensor networks, radio access technologies, communication protocols and complex systems.

R. Alcarria received his MS and PhD degrees in telecommunication engineering from the Technical University of Madrid, in 2008 and 2013, respectively. Currently, he is an associate professor at the Department of Geospatial Engineering of the Technical University of Madrid. He has been involved in several European and National R&D projects related to Future Internet, Internet of Things and service composition. His research interests are service architectures, sensor networks, human-computer interaction and prosumer environments.

T. Robles received a MS and PhD degrees in telecommunication engineering from Technical University of Madrid, in 1987 and 1991, respectively. He is a full professor of telematics engineering at the E.T.S.I. Telecommunication of the Technical University of Madrid. His research interest is focused on advanced applications and services for wireless networks, also on blockchain-based infrastructures.