A switching mechanism framework for optimal coupling of predictive scheduling and reactive control in manufacturing hybrid control architectures

Jose-Fernando Jimenez, Abdelghani Bekrar, Damien Trentesaux & Paulo Leitão

Published online: 02 May 2016.

Submit your article to this journal ⬈

View related articles ⬈

View Crossmark data ⬈

Taylor & Francis
Taylor & Francis Group

A switching mechanism framework for optimal coupling of predictive scheduling and reactive control in manufacturing hybrid control architectures

Jose-Fernando Jimenez[a,b*] ⓘ, Abdelghani Bekrar[a], Damien Trentesaux[a] and Paulo Leitão[c,d]

[a]*LAMIH, UMR CNRS 8201, University of Valenciennes and Hainaut-Cambrésis, UVHC, le Mont Houy, Valenciennes, France;*
[b]*Industrial Engineering Department, Pontificia Universidad Javeriana, Bogotá, Colombia;* [c]*Department of Electrical Engineering, Polytechnic Institute of Bragança, Campus Santa Apolónia, Bragança, Portugal;* [d]*LIACC – Artificial Intelligence and Computer Science Laboratory, Porto, Portugal*

Nowadays, manufacturing systems are seeking control architectures that offer efficient production performance and reactivity to disruptive events. Dynamic hybrid control architectures are a promising approach as they are not only able to switch dynamically between hierarchical, heterarchical and semi-heterarchical structures, they can also switch the level of coupling between predictive scheduling and reactive control techniques. However, few approaches address an efficient switching process in terms of structure and coupling. This paper presents a switching mechanism framework in dynamic hybrid control architectures, which exploits the advantages of hierarchical manufacturing scheduling systems and heterarchical manufacturing execution systems, and also mitigates the respective reactivity and optimality drawbacks. The main feature in this framework is that it monitors the system dynamics online and shifts between different operating modes to attain the most suitable production control strategy. The experiments were carried out in an emulation of a real manufacturing system to illustrate the benefits of including a switching mechanism in simulated scenarios. The results show that the switching mechanism improves response to disruptions in a global performance indicator as it permits to select the best alternative from several operating modes.

Keywords: hybrid production system; distributed manufacturing control; flexible job shop; production planning; agent-based systems; dynamic scheduling; predictive/reactive optimal coupling; switching mechanism

1. Introduction

Traditionally, manufacturing industries have a rigid and sequential business process for planning and production control. The production control problem is largely allocated to hierarchical decisional entities that perform sequential planning, scheduling and execution control of manufacturing operations (i.e. material resource planning and enterprise resource planning). This classical and first approach is based on a hierarchical architecture that uses a predictive scheduling system (often mathematical-programming or metaheuristics techniques) to achieve an optimised production plan. The drawbacks of this decisional arrangement are that the system struggles to provide optimal solutions in reasonable time and to react to environmental disturbances (Borangiu et al. 2015).

Since 1990s, due to increasing internal (breakdowns, etc.) and external (evolution of market demand, supply shortages, etc.) unpredictability, combined with the increasing complexity of manufacturing processes, more decision-distributed approaches have been considered in manufacturing control systems (Trentesaux 2009). In contrast to the previous approach, the production control is distributed among cooperative, autonomous and communicative decisional entities that execute operations and respond to environmental disturbances (Dilts, Boyd, and Whorms 1991). This second approach is based on a heterarchical architecture that uses reactive production control to achieve responsiveness on the manufacturing shop floor through cooperative decisional entities (i.e. agents, holons and intelligent products). However, the drawback is that it does not ensure optimal production performance and it is hard to demonstrate the minimal level of performance under normal conditions. Nonetheless, some effort has been made to solve these issues (Pach et al. 2012; Borangiu et al. 2014; Lee and Prabhu 2014).

Thus, over the last decade, researchers have developed a third approach that integrates the benefits of hierarchical and heterarchical architectures and mitigates the drawbacks. This approach, named hybrid control architectures (HCA) or semi-heterarchical architectures, integrates hierarchical and heterarchical architectures. It aims simultaneously to

---

*Corresponding author. Email: j-jimenez@javeriana.edu.co

achieve optimal production scheduling (called 'predictive scheduling' in this paper) and to react to unexpected events (called 'reactive control' in this paper). There are thus different possibilities to combine hierarchical and heterarchical organisational arrangements to obtain an integrated HCA. Moreover, this combination can evolve with time, leading to the dynamic evolution of the HCA architecture, such as it may evolve towards a more hierarchical architecture, a more heterarchical architecture or a more balanced architecture, for example. In this paper, if the combined HCA architecture is defined from the outset, whatever happens during execution, it is called 'static' (denoted S-HCA). On the contrary, if its architecture can be modified in response to certain events, it is called 'dynamic' (denoted D-HCA).

In a previous study on a specific manufacturing case, it was shown that D-HCAs tend to perform better than S-HCA (Pach et al. 2014). The internal mechanism of D-HCA enables their architecture to evolve over time in response to unexpected events. Despite its local scope of availability, this previous study showed the potential benefits of D-HCA. For example, if there is no perturbation, a D-HCA can retain a hierarchical organisation to ensure schedule optimality. However, the D-HCA can 'switch' to a fully heterarchical architecture, or even a mixed hierarchical–heterarchical architecture, if a perturbation occurs in order to react to this perturbation. As we will point out in the following section, there is little research in the literature on this switching mechanism, which is the topic of the paper. More precisely, we suggest that the evolution of the D-HCA is governed by the search for optimal coupling between predictive scheduling and reactive control mechanisms. From our point of view, optimal coupling is a specific configuration of D-HCA where it maximises (or minimises) the value of certain performance indicators (e.g. minimising makespan and/or mean square deviation). Therefore, it is assumed in this paper that D-HCA has the capacity to switch architecture through what we define as different operating modes. An operating mode is a specific parameterisation (definitions of all parameters) that characterises the concurrent setting of the D-HCA (Jimenez et al. 2015a). Switching (in the architecture) is the action of changing from one operating mode to another one following expected or unexpected events. The gradual evolution towards hierarchy in the D-HCA represents the will to use more global, optimal and predictive decisional techniques (typically, predictive scheduling algorithms), while the gradual evolution towards heterarchy in the D-HCA represents the will to use more local and reactive decisional techniques (typically, reactive control algorithms). Figure 1 illustrates the concept of switching operating modes to obtain an optimal coupling. Therefore, the challenge of D-HCA is to determine an adequate switching mechanism that makes the architecture evolve through different operating modes, to achieve an optimal coupling.

Three generic issues regarding this challenge in D-HCA are specifically addressed in this paper. When should the switching of the operating mode be performed? How should the switching of the operating mode be executed? Which actions are required to synchronise the new operating mode properly? In this paper, we will refer to these issues as
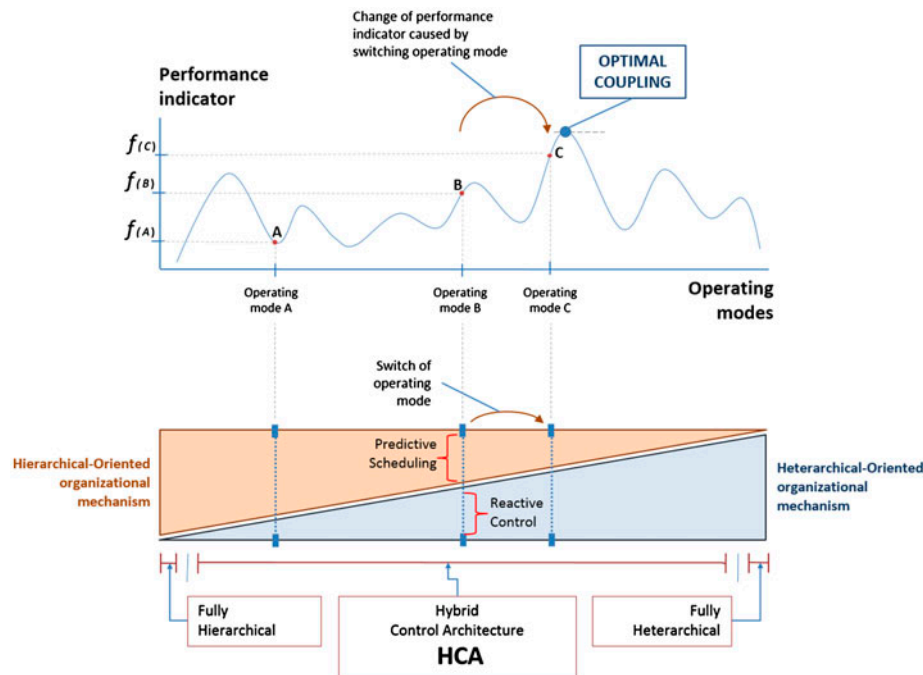

Figure 1.   Concept of switching operating modes to attain optimal coupling (Jimenez et al. 2015b).

switching timing, switching technique and switching synchronisation, respectively. From our point of view, an optimal coupling between the predictive and reactive techniques is achieved by defining a clear and integrated mechanism that addresses simultaneously these issues.

Accordingly, this paper addresses these issues and proposes a switching mechanism framework that searches for an optimal coupling of predictive scheduling and reactive control in D-HCA. In this paper, we conduct a preliminary study of our research project on the inclusion of a switching mechanism and its potential benefits. The paper is organised as follows. In Section 2, the approaches for optimising the coupling of predictive scheduling and reactive control technologies are reviewed. The proposed D-HCA that includes the switching mechanism framework is described in Section 3. Section 4 describes the application of the proposed approach to a real flexible manufacturing system and presents the experimental case study and results. Finally, Section 5 rounds-up the paper with our conclusions and some prospects for future research.

2. Literature review

In this section, we review the literature in the domain of D-HCA applied to manufacturing with a switching mechanism as a method of achieving optimal coupling in the system architecture (Figure 2). The papers reviewed have been studied according to the way they address the three issues introduced.

For the switching timing issue, three different approaches were identified according to disturbance detection and the triggering of the switching mechanism. In the first approach, agents trigger switching locally (local detection in Figure 2). The agents initiate a heterarchical-based solution due to their inability to execute their own objectives (Valckenaers et al. 2007; Novas et al. 2013; Herrera, Thomas, and Parada 2014). The advantage of this solution is that in cases where perturbations cause degradation in efficiency (not disruptions caused by breakdowns), the distributed entities might only require minor architectural changes and might recover autonomously. However, the drawback is that switching is limited and it may eventually not be sufficient to respond to major disruptions. In the second approach, a supervisor/coordinator activates the switching mechanism as it has a global view of the system (global detection in Figure 2). This supervisor monitors the system execution and/or system dynamic to activate the triggering (Raileanu et al. 2012; Borangiu et al. 2015). The advantage of this approach is that the detection might trigger structural and/or behavioural switching due the visibility of the entire system's behaviour. However, the drawback is that these approaches require a huge amount of computing time that may delay detection of the disturbance. In addition to
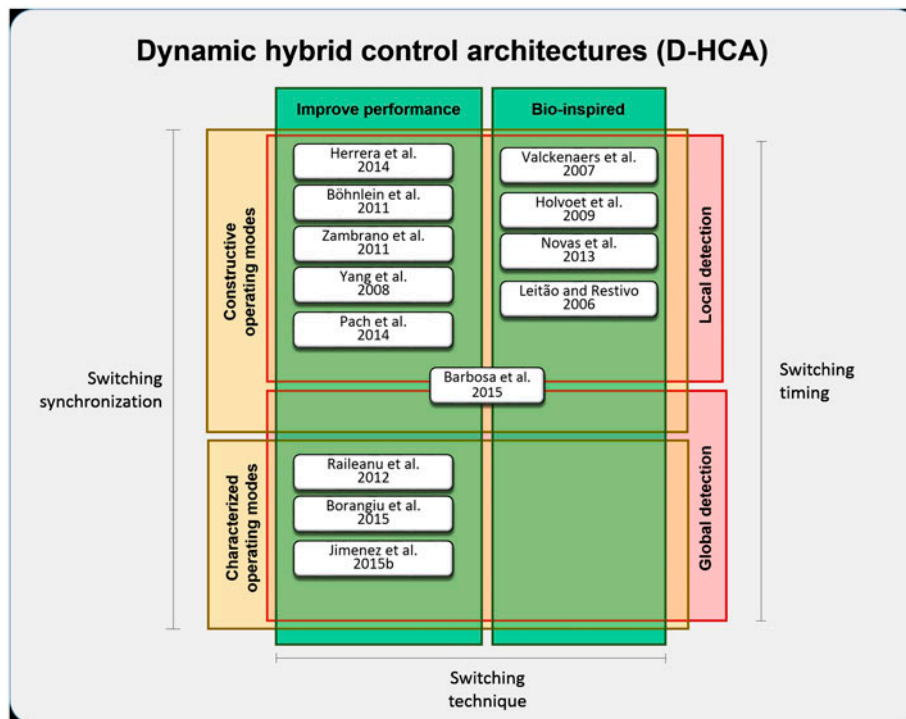

Figure 2. Classification of dynamic hybrid control architectures in manufacturing systems.

previous approaches, a third intermediary approach considers that the switching is activated either by local or global entities. In this case, two disturbance detection mechanisms are created to trigger micro or macro self-organisation (Barbosa et al. 2015). The result of this new approach is that the switch is performed according to the type of disruption.

Regarding the switching technique issue, we identified two different approaches: switching based on bio-inspired technologies and switching based on performance improvement. The bio-inspired approaches exploit agent coordination to gain a self-organisation characteristic (bio-inspired in Figure 2). The advantage is that the myopia is mitigated, as the product agents receive either knowledge about the current conditions with pheromones trails as in ADACOR (Leitão and Restivo 2006); or acknowledgement of possible future conditions with exploratory ants as in D-MAS (Holvoet, Weyns, and Valckenaers 2009). However, the disadvantage is that this approach does not guarantee an optimal solution and it does not attain an optimal operating mode. The performance improvement approach (improve performance in Figure 2) is based on the comparison between a system performance indicator and an expected control reference or threshold (Jimenez et al. 2015a). In this approach, a switching mechanism in dynamic hybrid control architectures was tested on very simple examples to illustrate its potential benefits. The advantage is that switching does not focus solely on achieving optimal production execution, but also on the behaviour of the dynamic system. The drawback is that the monitoring mechanism is a centralised functionality where a large amount of information is needed (Le Mortellec et al. 2013). Moreover, if the observation process is only activated when the threshold is disrupted, the system will not be able to support out-of-control tendencies.

The switching synchronisation issue depends on knowledge of the new operating mode. In this respect, we identified two approaches according to the characteristics of the operating mode: constructive-based and characterised-based operating modes. The system with a constructive-based operating mode (Figure 2) evolves to a better configuration to respond to corresponding necessities (Yang et al. 2008; Zambrano et al. 2011). In fact, the system constructs the new operating mode in response to disruptive events. The ORCA architecture (Pach et al. 2014) is a good example of such approach, as it switches its structural arrangement from a hierarchical architecture to a semi-heterarchical architecture due to jobs gradually changing their behaviour when a perturbation occurs. The main advantage of this approach is that it features continuous evolution in terms of allowing straightforward synchronisation after the switch. However, this approach does not demonstrate clear identification of the operating mode due to the difficulty of knowing in advance the expected configuration after switching. Characterised-based approaches (Figure 2) define a unique composition of each operating mode to be used by the switching mechanism (Raileanu et al. 2012; Borangiu et al. 2015; Jimenez et al. 2015b). This distinctiveness is interesting as it identifies the operating mode and allows the switching mechanism to choose from a variety of alternatives. The advantage is that it shows in advance an expected objective of the current manufacturing execution associated with each operating mode based on knowledge of the control configuration before and after switching. However, online synchronisation is difficult due to the complexity of making changes to the agents' intentions.

In general, the approaches in the literature reviewed focus on ensuring the continuity of production execution regardless of performance optimisation. As mentioned in Cardin et al. (2015), the shortcoming of these approaches is centred on the way the coupling of prediction and reaction is neither optimised nor justified. In this respect, we think that there are three limitations in the existing literature. Firstly, a clear definition of operating modes that gives the flexibility to choose different directions in the switching is lacking. A switching mechanism should have alternatives with different outcomes in order to adjust efficiently to the control system needs. Secondly, the lack of optimal coupling is caused by the absence of a mechanism that communicates beforehand the possible or expected behaviour of the system. Information regarding the new possible outcome is necessary to make decisions in the switching mechanism and to avoid uncertainty. Finally, the optimisation techniques used to search for the best configuration are based on bio-inspired and local search methodologies. It is worth exploring different techniques such as metaheuristics or combinatorial optimisation for the switching mechanism in order to seek the highest achievable performance. For this reason, the literature review reveals that there is a need for research on the coupling featured in D-HCA focusing on the acknowledgement of the expected behaviour of the operating modes and the combined effect derived from the switching timing, the switching technique and the switching synchronisation. This research deals with orchestrating these issues as a contribution in this field. The next section describes a general framework of a D-HCA in a manufacturing context that can support a generic switching mechanism aiming to reach the suggested optimal coupling of predictive scheduling with reactive control.

3. Proposal

3.1 *A D-HCA with a switching mechanism*

This section describes the generic D-HCA that includes the proposed switching mechanism. This paper aims to generalise and conceptualise the initial work presented in Jimenez et al. (2015b). It seeks to provide a framework to construct
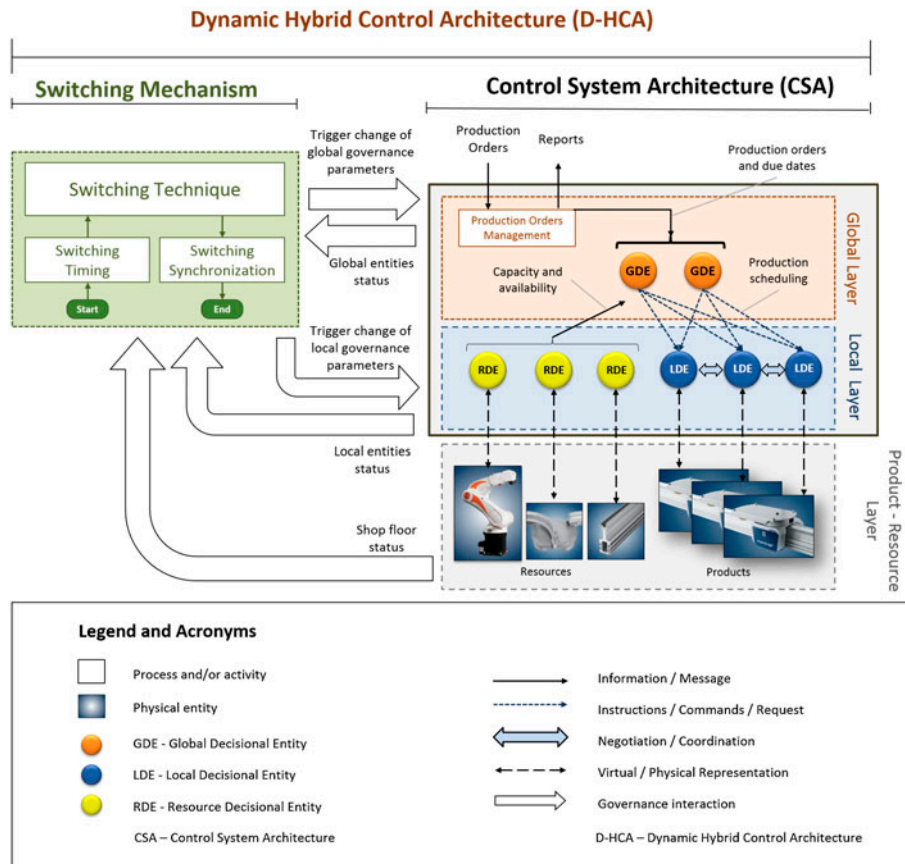
Figure 3. The D-HCA with the switching mechanism.

an efficient switching mechanism in dynamic hybrid control architectures while it creates a reference model to compare different switching approaches to handle this mechanism.

The proposed D-HCA, which includes a switching mechanism, is provided in Figure 3. The D-HCA is divided conceptually into a control system architecture (CSA) and a switching mechanism. The CSA has three different layers: a global layer, a local layer and a product/resource layer. The global layer provides the total planning of a production order (i.e. scheduling, dispatching, sequencing and routing) for a long-term period to optimise a global production objective. The local layer manages the online execution of the production order (i.e. dispatching planned orders, sequencing and routing of work-in-progress) with the capability of reacting to disruptions according to local execution objectives. The product/resource layer hosts the low-level control of the physical entities. The switching mechanism consists of three sequential components that are activated during each switching event. A switching timing component, which monitors the CSA and triggers switching when required; a switching technique component, which searches for an optimal operating mode; and a switching synchronisation component, which applies the new operating mode to the CSA. The next section details the functioning of these mechanisms.

Decisional entities in the global/local layer and physical entities in the product/resource layer compose the architecture. The decisional entities can be modelled using either holonic or multi-agent paradigms, for example, intelligent agents based on goal-oriented agents (Russel and Norvig 2003). A global decisional entity (GDE) and a local decisional entity (LDE) are located in the global or local layer, respectively. These entities function in an iterative algorithm for decision-making. The local layer also has a Resource Decisional Entity (RDE), which is created to control resources. Each entity starts by sensing the production system state via the communication component. Then, with the aim of executing the entity's goal (e.g. minimising makespan or choosing the shortest path), the decision-making technique (i.e. condition action rules, fuzzy logic rules, heuristics or optimisation techniques) is activated subject to the current entity configuration parameters. In this approach, the agent configuration parameters are called governance parameters. These governance parameters contain the attributes and rules of conduct that dictate the entity behavioural guidelines. Once the decision-making technique defines its actions to achieve the goals, the actions are implemented by its executing
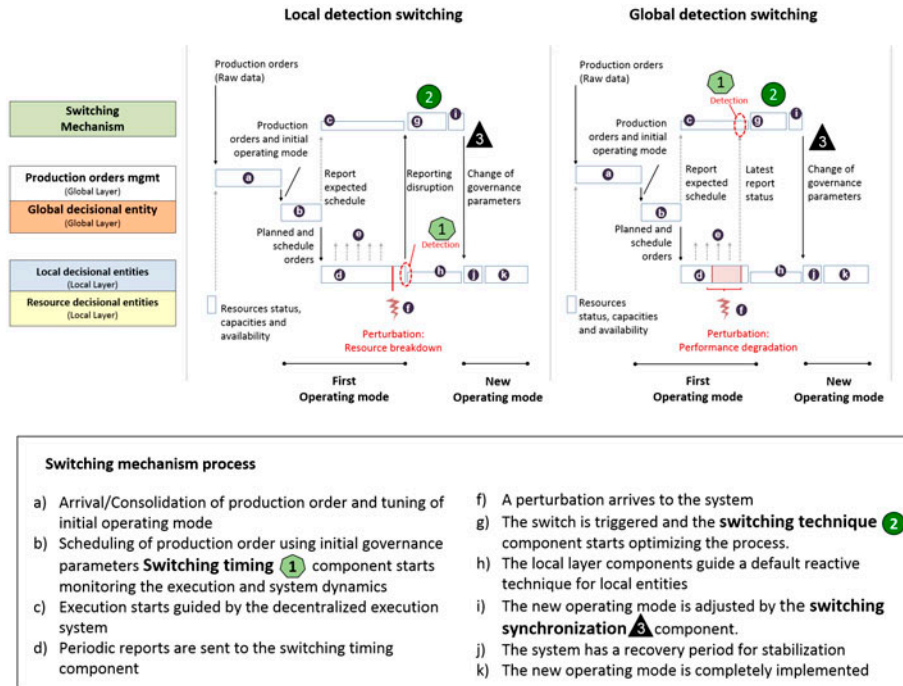
Figure 4. General process of the switching mechanism included in the D-HCA.

components (autonomous actions in the local layer) or transmitted by its communication components (command actions in the global layer). During the entire process, a data storage component collects the relevant information from the system and the entities' behaviour. The physical entities in the product/resource layer are real tangible entities such as parts or products guided by LDE, or conveying components or robots guided by RDE.

3.2 Switching mechanism framework

In this paper, the switching mechanism focuses on changing the configuration of the CSA. With the aim of easing the representation of a specific configuration, it is created an operating mode to gather all governance parameters of the system. Therefore, an operating mode describes the configuration of a CSA and symbolises the control strategy for a given problem. For this purpose, the switching mechanism aims to find an optimal coupling represented in the operating mode by changing the settings of the corresponding governance parameters. The general process of the switching mechanism included in the D-HCA is illustrated in Figure 4.

Aligned with the three issues that challenge the D-HCA (namely when, how and which questions), the three components of the switching mechanism, switching timing, switching technique and switching synchronisation, are detailed:

(1) Switching timing component: this component is responsible for monitoring the execution and dynamics of the CSA in order to activate the switching. The switch can be triggered by an adverse event such as deterioration in performance, or by a positive intention such as improvement in performance of system execution. This component can use any analytics to enhance the monitoring process, as it needs to discover meaningful patterns in the monitored data and communicate them to the whole system (reporting, analysing, forecasting and/or predicting). The inputs of this component are the current status of the dynamic system and possible recommendations through feedback from previous switching timing executions. The outputs are the activation of the next procedure (moment of switching) and the information resulting from the analytics performed. Figure 5 presents the pseudo-code of the switching timing algorithm.

(2) Switching technique component: this component is defined as the method that the mechanism uses to change operating mode. With the same objective as an optimisation process (Blum and Roli 2003), this component guides the search process according to different possible solutions. In fact, the mechanism finds the optimal solution from this exploration. Some examples of this optimisation method are bio-inspired mechanism, meta-heuristics, combinatorial optimisation, neural networks and reinforcement learning, among others. The inputs of

Algorithm 1: Switching timing algorithm

    Step 0: Set-up of algorithm
1:    Set $i \leftarrow 0$
2:    Set type of switching timing for the switching mechanism for global and/or local detection
3:    Set the monitoring times of algorithm $t_0, t_1, \ldots, t_{max}$
4:    Set time of execution $t \leftarrow t_i$
5:    Declare variables for statistical data S
6:    Declare state variables X of manufacturing cell
7:    Declare function for local detection to compute desired performance indicator of execution $f(t,X)$
8:    Declare function for global detection to compute desired performance indicator of system dynamics $g(t,X,S)$
9:    Set thresholds α_1 and β_1 for minimum and/or maximum tolerance for $f(t,X_t)$
10:    Set thresholds α_2 and β_2 for minimum and/or maximum tolerance for $g(t,X,S)$
11:    Declare function for updating statistical data $h(t,X,S)$
    Step 1: Initialization
12:    Wait until production execution starts
    Step 2: Stopping
13:    If $t = t_{max}$
14:    Stop algorithm
15:    If execution finished
16:    Stop algorithm
    Step 3: Gathering.
17:    Retrieve current state of manufacturing cell x_t
18:    Assign current state to state variables $X \leftarrow x_t$
    Step 4: Evaluation for global and local detection
19:    Compute $f(t,X)$ and $g(t,X,S)$
    Step 5a: Switching by local detection
20:    If $\alpha_1 \leq f(t,X) \leq \beta_1$
21:    Go to step 6
22:    Else,
23:    Activate switch
24:    Send time t and Statistical data S to *switching technique algorithm*
25:    Stop algorithm
    Step 5b: Switching by global detection
26:    If $\alpha_2 \leq g(t,X_t,S) \leq \beta_2$
27:    Go to step 6
28:    Else,
29:    Activate switch
30:    Send time t and Statistical data S to *switching technique algorithm*
31:    Stop algorithm
    Step 6: Advance to next monitoring point
32:    Update statistical data $S \leftarrow h(t,X,S)$
33:    Set $i \leftarrow i + 1$
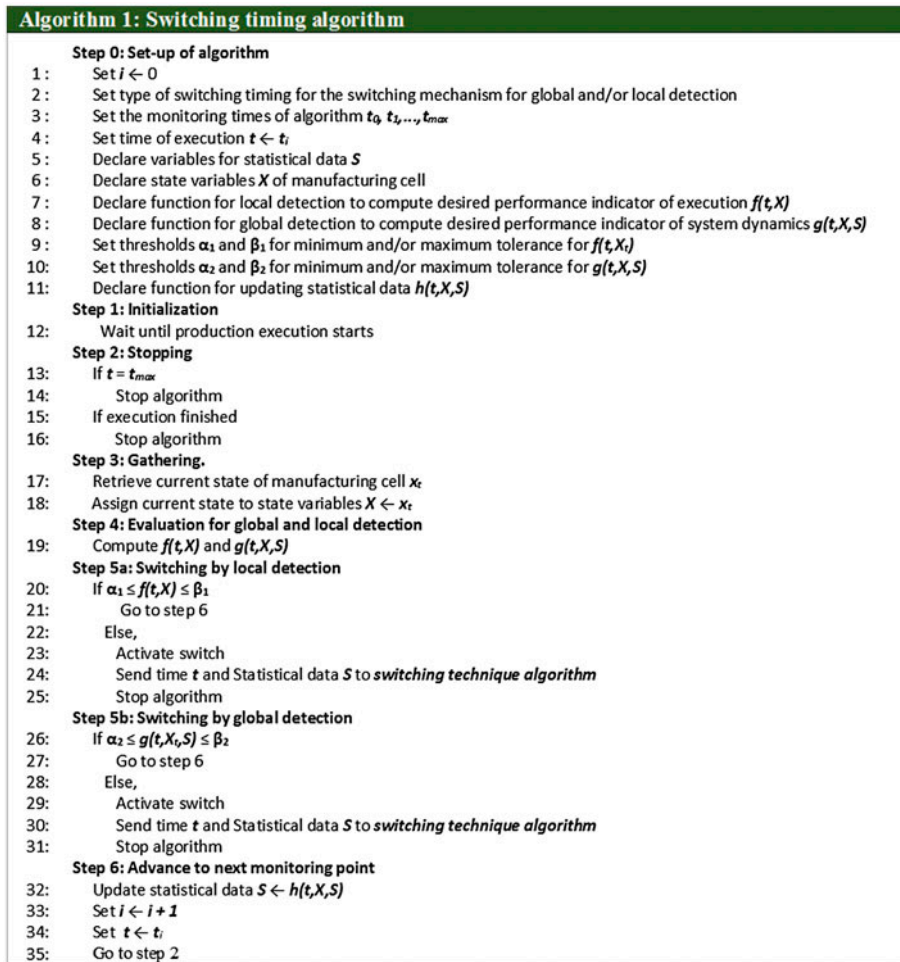34:   Set $t \leftarrow t_i$
35:    Go to step 2

Figure 5. Algorithm for the switching timing component.

this component are the current status of the dynamic system, the moment of switching according to the previous stage, the results of the analytics techniques and possible prior experiences of the switching technique. The outputs are the information from the guided search and the new operating mode to be implemented. Figure 6 presents the pseudo-code of the switching technique algorithm.

(3) Switching synchronisation component: this component implements the new operating mode according to the implementation strategies defined. It manages the execution of the switch as it adjusts the operating mode to ensure feasible implementation. Here, it is necessary to monitor system updates online to obtain the actual current state and to refine the changes in order to guarantee switching feasibility. The inputs of this component are the new operating mode characteristics, the information from analytics, the current state of the system and the switching implementation strategies. The outputs are the instructions to be followed to implement the new operating mode. Figure 7 presents the pseudo-code of the switching synchronisation algorithm.

In this paper, an experimental case study based on an instantiation of a real flexible manufacturing system is presented. Still, even though the scope of this paper is manufacturing, this approach can be used in other fields such as logistics and health care systems, or wherever managers seek a dynamic architecture for a system that couples predictive scheduling and reactive control in its execution. The following part details this application.

4. Application to the AIP PRIMECA flexible manufacturing system

The proposed approach will be applied to an instantiation of a real flexible manufacturing system, located at the Université de Valenciennes et Hainaut-Cambrésis, called AIP PRIMECA. The AIP PRIMECA can, theoretically, be seen as a
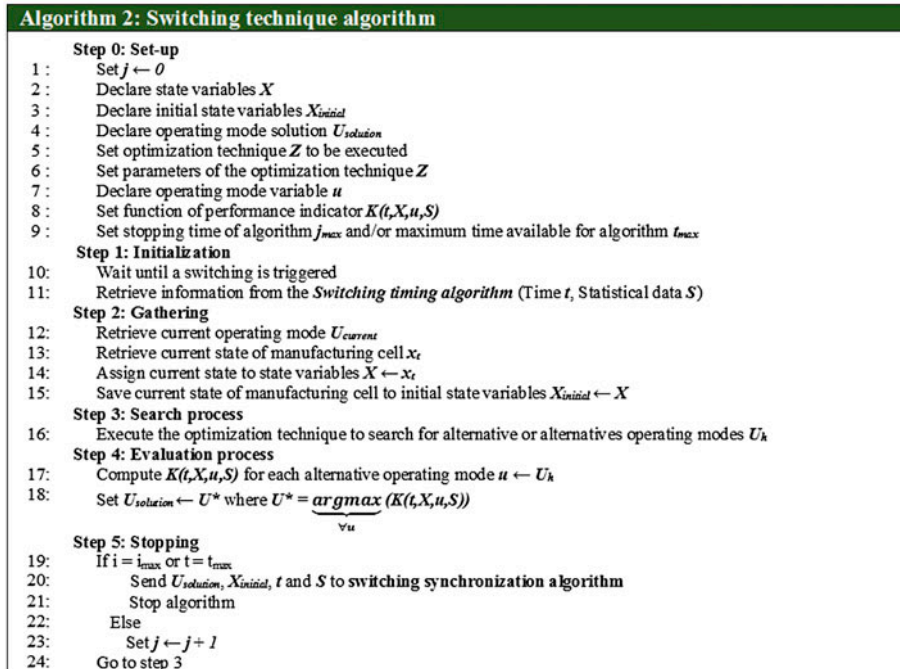
Algorithm 2: Switching technique algorithm

Step 0: Set-up
1 : Set $j \leftarrow 0$
2 : Declare state variables X
3 : Declare initial state variables $X_{initial}$
4 : Declare operating mode solution $U_{solution}$
5 : Set optimization technique Z to be executed
6 : Set parameters of the optimization technique Z
7 : Declare operating mode variable u
8 : Set function of performance indicator $K(t,X,u,S)$
9 : Set stopping time of algorithm j_{max} and/or maximum time available for algorithm t_{max}

Step 1: Initialization
10: Wait until a switching is triggered
11: Retrieve information from the *Switching timing algorithm* (Time t, Statistical data S)

Step 2: Gathering
12: Retrieve current operating mode $U_{current}$
13: Retrieve current state of manufacturing cell x_t
14: Assign current state to state variables $X \leftarrow x_t$
15: Save current state of manufacturing cell to initial state variables $X_{initial} \leftarrow X$

Step 3: Search process
16: Execute the optimization technique to search for alternative or alternatives operating modes U_k

Step 4: Evaluation process
17: Compute $K(t,X,u,S)$ for each alternative operating mode $u \leftarrow U_k$
18: Set $U_{solution} \leftarrow U^*$ where $U^* = \underset{\forall u}{argmax}\ (K(t,X,u,S))$

Step 5: Stopping
19: If $i = i_{max}$ or $t = t_{max}$
20: Send $U_{solution}, X_{initial}, t$ and S to switching synchronization algorithm
21: Stop algorithm
22: Else
23: Set $j \leftarrow j + 1$
24: Go to step 3

Figure 6.　Algorithm for the switching technique component.

Algorithm 3: Switching synchronization algorithm

Step 0: Set up
1 : Set synchronization technique W to be executed
2 : Set synchronization technique
3 : Declare state variables X
4 : Declare synchronization changes ΔX

Step 1: Initialization
5 : Wait until
6 : Retrieve information from the *Switching technique algorithm* (operating mode solution $U_{solution}$, initial state variables $X_{initial}$, Time t, statistical data S)

Step 2: Gathering
7 : Retrieve current state of manufacturing cell x_t
8 : Assign to state variables $X \leftarrow x_t$

Step 3: Evaluation of synchronization
9 : Compute differences between current state X and initial of switching $X_{initial}$

Step 4: Considerations for synchronization
10: For each difference between of states
11: Calculate the synchronization changes ΔX by the synchronization technique W

Step 5: Implement operating mode
12: Set new operating mode $U_{solution}$ in system and consider the synchronization changes ΔX
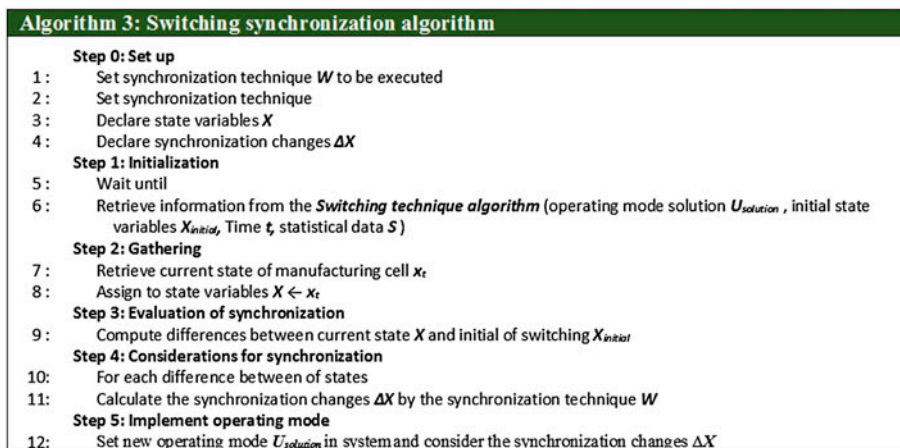
Figure 7.　Algorithm for the switching synchronisation component.

flexible job shop, leading to the well-known flexible job-shop problem (FJSP) in the operation research literature. In this part, the proposed D-HCA was applied to a general FJSP before being applied to the AIP PRIMECA Flexible manufacturing system. Afterwards, the experimental protocol and the results are presented.

4.1 *Switching mechanism framework*

A flexible job-shop problem (FJSP) is a classical optimisation problem in scheduling in which a set of jobs composed of a sequential collection of operations need to be assigned to the available resources in a flexible manufacturing system. This problem consists mainly in solving simultaneously two subproblems: job routing, as each operation of each job to be executed needs to be assigned to one of the available machines, and the scheduling subproblem, as the operations assigned to all machines need to be sequenced in order to obtain a feasible solution (Fattahi and Fallahi 2010).

Also known as an extension of the job-shop problem that requires the selection of machines for the execution of jobs (Brandimarte 1993), the FJSP is an example of a strong NP-Hard problem due to its complexity (Pinedo 2012). A general formulation of a FJSP that includes transport times and limited job capacity is presented as follows.

The FJSP can be defined as the allocation of n jobs to m machines connected in a transportation system with recirculation (Pinedo 2012). The objective of solving this problem is to minimise the makespan that results from processing all jobs. Each job requires a number of sequenced operations Oij where operation j can be performed by the machine i. The problem to be solved is the dispatching, machine sequence and product routing of the jobs through the flexible manufacturing system. A complete mathematical formulation of the FJSP can be found in Trentesaux et al. (2013).

For the FJSP, the proposed D-HCA with a switching mechanism can be instantiated as follows:

- Global layer: this layer might contain one or many global decisional entities (GDEs) with a goal that concerns global performance such as the order completion time (makespan). The decision-making technique for each GDE can be any optimisation model that resolves the FJSP in order to provide dispatching, machine sequence and routing.
- Local layer: this layer contains as many local decisional entities as jobs to be produced in the manufacturing cell. Each LDE is assigned a goal in order to follow in case it has no imposed objectives (e.g. minimise completion of next operation). Dispatching rules, reactive techniques (i.e. contract net or potential fields) or heuristics can be used to guide dispatching, machine sequence and routing in the manufacturing system. RDE are machines, conveyors and gates (resources) that might interact in the layer.
- Operating modes: the operating mode vector represents the production and control strategies, which are changed by the switching mechanism. This vector gathers the governance parameters from all the global and local decisional entities (GDE and LDE). The number of operating modes results from the combinatorial possibilities within the different governance parameters. In addition to the vector, each operating mode has a general fitness that varies according to the expected result in the dynamic system. The general fitness is a unique characterisation function that evaluates the quality of the operating mode (Jimenez et al. 2015a). This fitness is the decisional criterion to decide whether to switch to a new operating mode.
- Switching mechanism: the three components, defined in Section 3.2, can be applied as follows. The switching timing relies on detecting perturbations in the FJSP by evaluating global and local indicators. On the one hand, the local indicator measures the performance regarding the completion of a local objective. This is located in each LDE, which represent the jobs in the FJSP. This indicator is modelled with the function $f(t, X)$ and contains a lower α_1 and an upper β_1 threshold, for which violation will activate the switching. On the other hand, the global indicator measures the performance regarding the completion of the global objective. This indicator is $g(t, X, S)$ and also contains a lower α_2 and an upper β_2 threshold. Yet, this detection function includes an analytical calculation that detects current or future objective failures. The switching technique relies on choosing an operating mode from the set of alternatives. This component considers any function as a mathematical or a simulation-based function. A mathematical calculation of the tardiness and a simulation of the production execution are both examples of this function. Switching synchronisation is a feasibility method to implement the new operating mode that solves the FJSP. In this case, it is defined as a heuristic that implements the operating mode in the CSA. This component depends on many factors such as limited switching time or complexity of implementation. Still, this synchronisation may be as simple or complex as desired. An example of a heuristic for the switching synchronisation can be a post-optimal metaheuristic that fixes the operating mode after switching. Nevertheless, its proposal should maintain or avoid major changes to the solution given by the switching technique.

4.2 *Experimental case study: application to the AIP-PRIMECA*

In this section, an instantiation of a real flexible manufacturing system is presented to validate the benefits of the proposed D-HCA. A simulation is used to solve an online FJSP with perturbations.
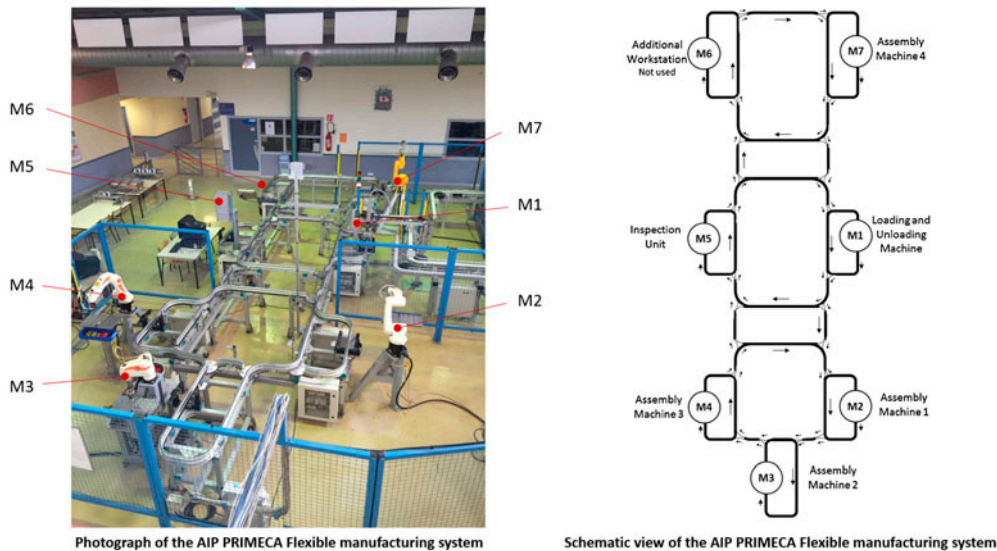
4.2.1 *AIP PRIMECA: instantiation of a FJSP*

The proposed manufacturing cell for this experimental case study is a simulation based on a real flexible manufacturing system. The AIP-PRIMECA can be formulated as a FJSP with recirculation. This facility is composed of seven workstations connected by a material handling machine. The workstations are a loading/unloading machine (M1), four assembly machines (M2, M3, M4 and M7), an inspection unit (M5) and an additional workstation (M6) not used in this paper. In the experimental case studied, seven types of jobs can be produced (Type B, E, L, T, A, I and P). Each type of job has a sequence of operations to be executed by the machines from a set of operations (O1, O2, O3, O4, O5, O6, O7 and

O8). The information concerning the operation sequence for each type of job, the operations processed by each machine, the processing times and the transportation times are presented in Figure 8. Further information about the AIP PRIMECA flexible manufacturing system is available in (Trentesaux et al. 2013).

Aligned with the design introduced previously for FJSP, the D-HCA was configured and implemented for the AIP PRIMECA:

- *Global layer*: a single GDE is instantiated with a mixed integer linear programme (MILP) for solving the dispatching and machine sequence of the production order. The objective of the GDE is to minimise the makespan of the production order. The GDE is programmed in JAVA as it can invoke the MILP model using the IBM ILOG CPLEX Optimizer package (IBM ILOG CPLEX 2015). The mathematical formulation of the MILP is presented in Trentesaux et al. (2013). The governance parameters (2) used by the GDE are related to its correspondent role towards the jobs' dispatching and machine sequence. The possible roles are either coercive or permissive. A coercive role of the GDE is when the entity imposes the intentions or decisions on the LDE following the MILP solution results (dispatching and machine sequence). The permissive role is when the entity does not take part in the decision-making of LDE decision.
- *Local layer*: as many LDEs as jobs in the production order are created. Each LDE is instantiated with two techniques for dispatching and a heuristic for machine sequence. The objective of each LDE is to minimise the completion of the subsequent operation. Consequently, the governance parameters (2) are the techniques used in the decision-making. For the dispatching, the possible values for the governance parameter are (a) dispatching rule 1: to adopt the decision-making technique as the LPT with the most operation changes in the process sequence, and (b) dispatching rule 2: to adopt the decision-making technique as the SPT with the fewest operation changes in the process sequence (Malstrom 1983). The second governance parameter is settled as a unique value as it uses only a potential fields approach for machine sequence allocation. A potential fields approach is used for machine

Photograph of the AIP PRIMECA Flexible manufacturing system

Schematic view of the AIP PRIMECA Flexible manufacturing system

Code	Letter	Operations Sequence
901	B	O7 - O1 - O1 - O1 - O2 - O2 - O5 - O3 - O6 - O8
902	E	O7 - O1 - O1 - O1 - O2 - O2 - O4 - O6 - O8
903	L	O7 - O1 - O1 - O1 - O5 - O5 - O3 - O3 - O6 - O8
904	T	O7 - O1 - O1 - O2 - O4 - O6 - O8
905	A	O7 - O1 - O1 - O1 - O2 - O4 - O5 - O3 - O6 - O8
906	I	O7 - O1 - O1 - O5 - O3 - O6 - O8
907	P	O7 - O1 - O1 - O2 - O4 - O6 - O8

Operation	Processing times per machine (Seconds)				
	M1	M2	M3	M4	M5
Operation 1 (O1)		20	20		
Operation 2 (O2)		20	20		
Operation 3 (O3)				20	
Operation 4 (O4)		20		20	
Operation 5 (O5)			20	20	
Operation 6 (O6)					5
Operation 7 (O7)	10				
Operation 8 (O8)	10				

Figure 8. AIP-PRIMECA Flexible manufacturing system (Topology, operations sequence, and processing times).

sequence allocation. The potential fields approach is a reactive technique that sets attraction and repelling forces to resources to guide the behaviour of entities (Pach et al. 2012). The RDEs control the resources and their role are static, as their behaviour is not changed by the switching mechanism.

- *Operating modes:* the operating modes of the CSA emerge from the combination of the governance parameters of the GDE and the LDEs. In this sense, the operating mode is a control vector of $2 + 2j$ elements, where j is the number of LDEs in the production order. An example of an operating mode for this experiment is:

$$\text{Operating mode1} = \{\text{Coercive, Coercive, Dispatching Rule1, Potential fields, ...}\} \tag{1}$$

where the first two elements of the vector are the governance parameters of the GDE for dispatching and machine sequence, and all the other elements are the governance parameters arranged in pairs according to the number of LDEs in the production order. In this experiment, the general fitness for each operating mode, which will provide the quality and indicator for the switching process, is calculated from the results of a simulation model of the AIP-PRIMECA developed using agent-based software, called Netlogo 5.2 (Wilensky 1999). In this instantiation, considering that all LDEs may have the same governance parameters, six operating modes are created randomly to evaluate the switching. Nevertheless, the diversity of the operating modes created emerges from the variability of the governance parameters of the GDE and the LDEs.

- *Switching mechanism:*

The switching timing defines two mechanisms to detect the perturbations and, consequently, trigger a switching. On one side, the global detection analyses the system by monitoring the average difference between the expected completion time and the execution time of the work-in-progress jobs. For the experimental case study, the global detection is defined as follows:

$$f(t, X) = \sum_{\forall j} (b_j - t) \geq \alpha \tag{2}$$

where b_j is the expected completion time of next operation derived from the predictive schedule, t is the current simulation time, X is the current state of the shop floor, and α is the minimum permitted threshold of system degradation. This equation measures the average earliness or tardiness of completion of the next operation. If this indicator exceeds the threshold, the switch is activated. It is recommended to set the α parameter to a negative value in order to permit degradation in the system up to a considerable limit. For the experiments case study, the parameter α is set to -10 for the global detection.

The local detection mechanism is responsible for detecting machine breakdowns. In this paper, local triggering is initiated by the LDE, which informs the switching mechanism that it is unable to achieve its objectives. In this case, once the information is received, switching is activated. For local detection, a detection mechanism is programmed in each entity that verifies the feasibility of performing the next operation. If it is not possible, the need to switch is communicated.

The switching technique is performed according to an improvement performance technique. At switching, the mechanism gathers information about the CSA and performs an iteration of a simulation-based optimisation to choose the new operating mode. The simulation is performed in parallel with the execution in a different instance of the same Netlogo Model. This model can be invoked from the JAVA programme. The parallel execution time for the order of seven jobs is 0.786 ms on average. In this sense, a switching technique simulating six operating modes would take approximately 4.71 s. The makespan of this parallel execution is the fitness of each operating mode vector.

Switching synchronisation is not used in this case study. The governance parameters do not specifically change the intentions of the jobs, but the governance parameters that dictate the behaviour of the entities are changed. For this, when the operating mode is switched, the entities function normally and define the intentions of the jobs when the new operating mode is applied. In this case, the synchronisation is not needed.

4.2.2 *Experimental protocol*

In this section, the experimental protocol is presented. The main goal is to evaluate the inclusion of the switching mechanism in a D-HCA and to test the effectiveness of the proposed model compared with other architectures. For the emulation, the experiment also used Netlogo to simulate the AIP PRIMECA flexible job shop. The experimental model has three variables: the data-set (production order) to process, the disruption type (resource failure) and the moment of the perturbation (disruption time).

The data-sets tested are called A0, B0 and C0. They define the production orders to be executed and are an instantiation of the data-sets extracted from the AIP PRIMECA benchmark (Trentesaux et al. 2013). Data-set A0 processes seven jobs (B, E, L, T, A, I, P) with no job limitations; data-set B0 processes seven jobs (B, E, L, T, A, I, P) with five jobs simultaneously; and data-set C0 processes seven jobs (B, E, L, T, A, I, P) with four jobs simultaneously. Another parameter tested in the experiment is the type of perturbation. This parameter, which defines the disrupted scenario, considers disruption 1 (DisR1), where machine M3 breaks down and disruption 2 (DisR2), where machine M3 increases the processing time of all operations by 100%. Finally, the last parameter is the moment of disruption, which considers the perturbation 22, 195 and 239 s after the start of execution and it lasts until all the jobs are processed. Considering all the parameters and combinations, this experiment has 18 different scenarios. Finally, the objective of the experiment was to compare the makespan of all the scenarios.

For the experiments, four different and concurrent control systems were defined to evaluate and compare the performance of the proposed D-HCA. The first three control systems were a D-HCA system with a switching mechanism. While CS1 is the approach proposed in this paper, CS2 and CS3 are based on the ORCA architecture (Pach et al. 2014), which is reviewed in the literature. We chose this architecture because we think that it is the closest to our proposal. The difference between CS2 and CS3 is that for CS2 all jobs switch their behaviour from a predictive scheduling technique to a potential field's approach, while for CS3 only perturbed jobs switch their behaviour from a predictive scheduling technique to a potential fields approach. CS4 is a S-HCA with a predictive-reactive approach providing a predictive technique for dispatching and a reactive technique for machine sequence. Finally, CS5 is a fully reactive, heterarchical architecture. These architectures are detailed as follows:

- CS1 – Control system 1 (Control system proposed in this paper)

  D-HCA approach with a switching mechanism that steers the operating mode to optimise predictive scheduling (MILP programme) and reactive control (SPT/LPT and potential fields) coupling.

- CS2 – Control system 2 (Pach et al. 2014)

  Modified ORCA approach that starts from a predictive schedule (MILP) and switches to reactive control (potential fields) for all the products in the manufacturing system (homogeneous switching) following a disruption.

- CS3 – Control system 3 (Pach et al. 2014)

  ORCA approach, like a D-HCA, that switches from a predictive schedule to reactive control for the products affected by the perturbation (heterogeneous switching).

- CS4 – Control system 4 (predictive-reactive approach)

  S-HCA approach that follows a predictive technique (MILP) for dispatching and a reactive technique (Potential fields) for machine sequence.

- CS5 – Control system 5 (fully reactive)

  Fully static and heterarchical reactive approach for production control with a reactive technique for dispatching (LPT-longest processing time) and machine sequence (Potential fields) of jobs.

To specifically evaluate the proposed D-HCA, two types of experiments were conducted. Firstly, Experiment A analyses the simulation results of the operating modes in the proposed model (CS1). Then, Experiment B compares the makespan between the five different architectures (CS1, CS2, CS3, CS4 and CS5) and presents a detailed comparative analysis of a particular case between our approach (CS1) and the ORCA approach with heterogeneous switching (CS3).

4.2.3 *Results*

The results of the experiments performed are presented in this section.

Figure 9 presents the makespan obtained from the simulation for each operating mode for the different scenarios. Firstly, the results reinforce that performance varies depending on the operating mode used in the control architecture. This variability confirms that the static hybrid control architecture (S-HCA) has reactive behaviour, but lacks any real optimality, as it does not choose from a set of alternatives. A switching mechanism that allows different alternatives to be chosen enhances the possibility of absorbing the degradations caused by disruptions. Secondly, these results show that, in addition, the consequences on overall performance are less detrimental if the perturbation occurs closer to the
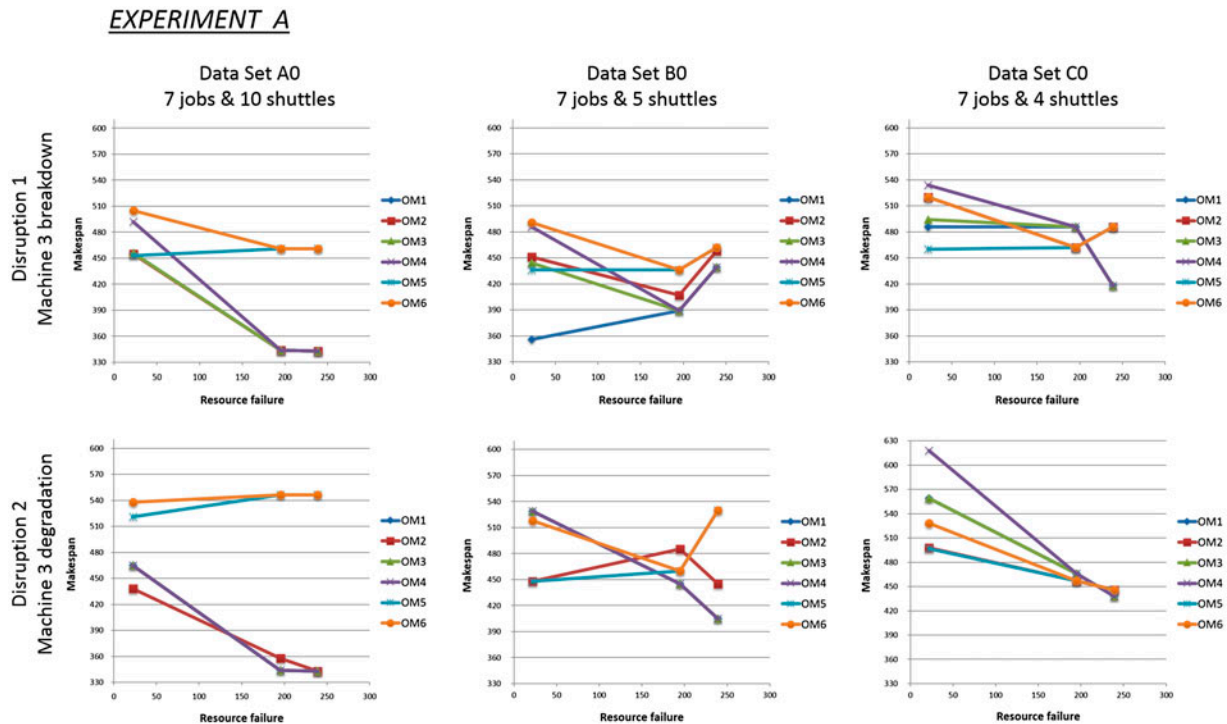
Figure 9. Results of the simulation of different operating modes in the proposed model (CS1).

end of production, as the different operating modes converge to the same performance. This situation occurs because once the system is closer to order completion it has already executed all the processes, leaving less flexibility and scope for the new execution configuration. On this matter, the switching mechanism should calculate the performance indicator of the operating modes by considering the time and/or progress of the production order. It is worth exploring mechanisms that acknowledge beforehand the possible results of different operating modes and use this as a decisional criterion for switching. The results demonstrate the contribution along these lines.

Figure 10a shows a comparison of the makespan of the different control systems. The results obtained show that, under the makespan criteria, the proposed approach (CS1) performs the best in six scenarios and shares the best performance in six scenarios. The CS2 approach is not the best in any scenario and shares the best performance in four scenarios. CS3 (ORCA) is the best in three scenarios and shares the best performance in five scenarios. CS4 and CS5 have the best performances in 2 and 1 of the 18 scenarios, respectively, while the proposed CS1 behaves better or equals the best for all the others. In this case, the results are justified by the fact that our CS1 model explores and chooses the best simulation from six different alternatives.

CS3 (ORCA) changes the machine sequence decision-making and follows the potential fields approach. Consequently, it moves the affected job (Letter L) from machine 3 to machine 7. The main problem with this approach is that it uses different resources that were allocated to different jobs, and thus the disruption indirectly affects the rest of the jobs. On the contrary, our model considers an operating mode that considers both affected and unaffected jobs. In the end, it reacts by optimising the situation of the work-in-progress. The degradation of the MILP solution is 101.03% for CS1 (our approach) compared with 108.57% for the CS3 (ORCA) approach. In this case, the good results are justified, as the switching technique explores six different alternatives.

To conclude from these simulations, it is worth mentioning that the inclusion of a switching mechanism in a D-HCA is beneficial for the agility needed in manufacturing systems. From the experiments performed, the switching mechanism benefits from the different alternatives derived from the operating modes to adjust its architecture to optimal coupling. In the other architectures, the switching is not optimal and only reacts to ensure the continuity of the execution. In addition, it is inferred that the optimal coupling is not a function just from the switching technique. In fact, to achieve an optimal coupling it is necessary to orchestrate the time of change (switching timing), the optimisation technique (switching technique) and the implementation of new control solution (switching synchronisation).

These preliminary findings in our research are promising. Meanwhile, we are aware that it is necessary to focus our next efforts to present a contribution that is easy to be adopted by the research community. As a consequence, the next

<u>EXPERIMENT B</u>

(a)

Data Sets	Predictive Makespan No perturbation	Reactive Makespan No perturbation	Experiment Scenario	Disruption Type	Time of failure	Makespan in Seconds				
						CS1 Proposed D-HCA	CS2 Modified ORCA Pach et al. 2014a	CS3 ORCA Pach et al. 2014a	CS4 Predictive Reactive no switching	CS5 Fully Reactive no switching
A0	343	370	S1	DisR-1	22	453	455	445	451	454
			S2		195	344	344	344	414	536
			S3		239	343	343	343	385	400
			S4	DisR-2	22	438	438	445*	428	468
			S5		195	344	358	343*	389	400
			S6		239	343	343	343*	370	400
B0	385	401	S7	DisR-1	22	436	451	451	451	454
			S8 **		195	389	407	418	442	448
			S9		239	439	458	458	442	445
			S10	DisR-2	22	448	448	429*	507	435
			S11		195	445	485	445*	425	452
			S12		239	405	445	405*	412	452
C0	418	492	S13	DisR-1	22	460	520	484	466	470
			S14		195	462	462	462	529	452
			S15		239	418	486	486	529	452
			S16	DisR-2	22	497	498	559*	529	498
			S17		195	457	457	466*	510	462
			S18		239	438	446	438*	492	457

*   The ORCA approach specifically features switching when machines break down. Results were obtained by imposing switching
     following machine degradation.
**   This scenario is detailed in the part B

(b)   **Scenario 8 –** Data set B0, type of disruption DisR-1, Time of failure 195
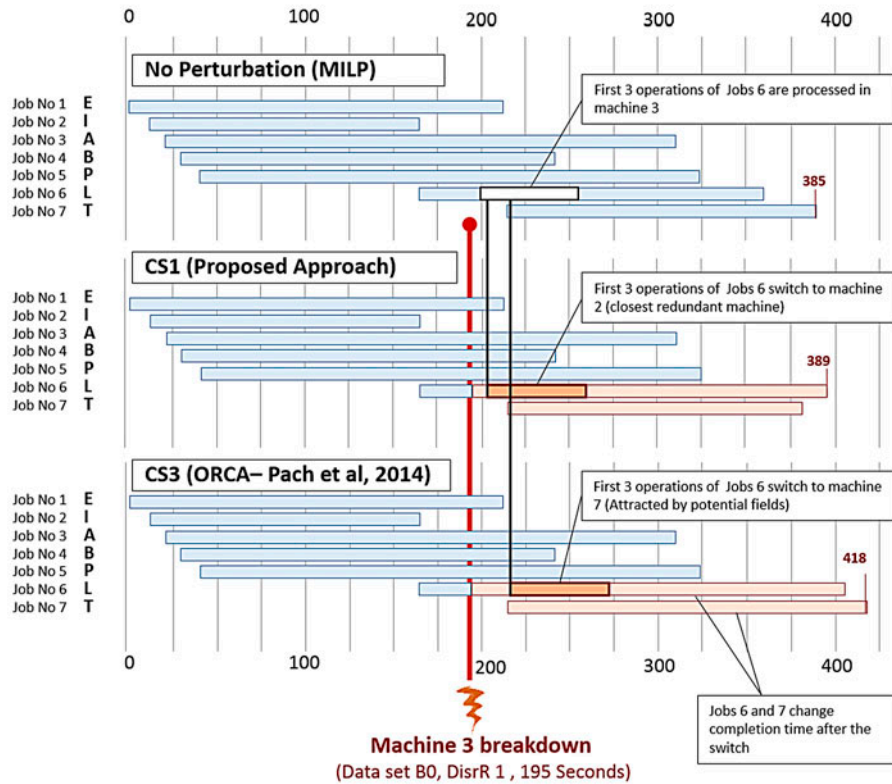

Figure 10.   (a) Makespan comparison for different control systems. (b) Results of the execution of scenario S8.

research steps are: (a) To formalise the proposed model in the UML modelling language to ease the researchers' appropriation and reproduction, (b) To construct a methodological guide to support the diverse applications of our contribution on other manufacturing systems and (c) To conduct real full-size experiments to evaluate rigorously our contribution and to improve the models through feedback improvement processes.

5. Conclusion and future work

In this paper, a switching mechanism framework that searches for optimal coupling of predictive scheduling and reactive control in D-HCA is proposed. Its role is to steer the operating mode of the CSA to attain the most suitable control solution. The result of the application of our framework to a manufacturing case study confirms that including a switching mechanism in a D-HCA enhances the agility for the optimality and reactivity of the system. The approach has some limitations related to the switching components. In the switching timing, it is possible that the mechanism frequently triggers the change without leaving a stabilisation period (leading to nervous behaviour, as defined by Barbosa et al. 2012). For this, it is recommended exploring a dampening mechanism to handle the frequency of switching. In the switching technique, depending on the number of alternatives and the complexity of the technique, a long time for switching might cause further degradation. It is worth analysing the balance between optimal coupling and the switching time to adopt a satisfactory solution.

There are three different lines of future research derived from the development of the components in the switching mechanism. First, the monitoring of the execution and system dynamics need to deploy real analytics. In this sense, this monitoring mechanism might not only deal with data reporting and current/trends analysis, but also forecasting and predicting possible behaviour of the system in the future. Once this is done, the switching mechanism will have better tools to choose the optimal operating mode. Second, in this paper, only six different operating modes were compared, with a total execution time of less than 5 s. In this sense, in order to support the combinatorial alternatives resulting from different governance parameters, and to avoid difficult synchronisation after the switch, it is necessary to develop a switching technique that compares different alternatives within a limited time. Finally, the resulting synchronisation issues need to be mitigated when the new operating mode is applied to the D-HCA. In this sense, the synchronisation might adjust the operating mode for implementation or include synchronisation variables in the switching techniques. This perspective allows a proper control policy to be established for a D-HCA and the integration of the switching concept to obtain the optimisation and reactivity required.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Jose-Fernando Jimenez http://orcid.org/0000-0001-8336-6240

References

Barbosa, J., P. Leitão, E. Adam, and D. Trentesaux. 2012. "Nervousness in Dynamic Self-Organized Holonic Multi-Agent Systems." In *Highlights on Practical Applications of Agents and Multi-Agent Systems* 9–17. Berlin Heidelberg: Springer.

Barbosa, J., P. Leitão, E. Adam, and D. Trentesaux. 2015. "Dynamic Self-Organization in Holonic Multi-Agent Manufacturing Systems: The ADACOR Evolution." *Computers in Industry* 66: 99–111.

Blum, C., and A. Roli. 2003. "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison." *ACM Computing Surveys (CSUR)* 35 (3): 268–308.

Böhnlein, D., K. Schweiger, and A. Tuma. 2011. "Multi-Agent-Based Transport Planning in the Newspaper Industry." *International Journal of Production Economics* 131 (1): 146–157.

Borangiu, T., S. Răileanu, D. Trentesaux, T. Berger, and I. Iacob. 2014. "Distributed Manufacturing Control with Extended CNP Interaction of Intelligent Products." *Journal of Intelligent Manufacturing* 25 (5): 1065–1075.

Borangiu, T., S. Răileanu, T. Berger, and D. Trentesaux. 2015. "Switching Mode Control Strategy in Manufacturing Execution Systems." *International Journal of Production Research* 53 (7): 1950–1963.

Brandimarte, Paolo. 1993. "Routing and Scheduling in a Flexible Job Shop by Tabu Search." *Annals of Operations Research* 41 (3, September): 157–183.

Cardin, O., D. Trentesaux, A. Thomas, P. Castagna, T. Berger, and H. B. El-Haouzi. 2015. "Coupling Predictive Scheduling and Reactive Control in Manufacturing Hybrid Control Architectures: State of the Art and Future Challenges." *Journal of Intelligent Manufacturing* 1–15. doi: 0.1007/s10845-015-1139-0

Dilts, D. M., N. P. Boyd, and H. H. Whorms. 1991. "The Evolution of Control Architectures for Automated Manufacturing Systems." *Journal of Manufacturing Systems* 10 (1): 79–93.

Fattahi, P., and A. Fallahi. 2010. "Dynamic Scheduling in Flexible Job Shop Systems by Considering Simultaneously Efficiency and Stability." *CIRP Journal of Manufacturing Science and Technology* 2 (2, January): 114–123.

Herrera, C., A. Thomas, and V. Parada. 2014. "A Product-Driven System Approach for Multilevel Decisions in Manufacturing Planning and Control." *Production & Manufacturing Research* 2 (1): 756–766.

Holvoet, T., D. Weyns, and P. Valckenaers, 2009. "Patterns of Delegate MAS". In *Self-Adaptive and Self-Organizing Systems, 2009. SASO'09. Third IEEE International Conference on IEEE*. September, 1–9.

Jimenez, J. F., A. Bekrar, D. Trentesaux, and P. Leitão. 2015a. "An Approach for Characterizing the Operating Modes in Dynamic Hybrid Control Architectures." In *Industrial Applications of Holonic and Multi-agent Systems*, edited by V. Marik, A. Schirrmann, D. Trentesaux, and P. Vrba. 108–119. Springer International Publishing. Valencia: Holomas. doi: 10.1007/978-3-319-22867-9_10

Jimenez, J. F., A. Bekrar, D. Trentesaux, G. Zambrano-Rey, and P. Leitão. 2015b. "Governance Mechanism in Control Architectures for Flexible Manufacturing Systems." *IFAC-PapersOnLine* 48 (3): 1093–1098.

Le Mortellec, A., J. Clarhaut, Y. Sallez, T. Berger, and D. Trentesaux. 2013. "Embedded Holonic Fault Diagnosis of Complex Transportation Systems." *Engineering Applications of Artificial Intelligence* 26 (1): 227–240.

Lee, J., and V. Prabhu. 2014. "Optimal Search of Distributed Arrival Time Control." *International Journal of Production Research* 53 (15): 4499–4510.

Leitão, P., and F. Restivo. 2006. "ADACOR: A Holonic Architecture for Agile and Adaptive Manufacturing Control." *Computers in Industry* 57 (2): 121–130.

Malstrom, E. M. 1983. "A Literature Review and Analysis Methodology for Traditional Scheduling Rules in a Flexible Manufacturing System", *Final Technical Report Performed under CAM-I Contract LASS-FM-07*.

Novas, J. M., J. Van Belle, B. Saint Germain, and P. Valckenaers. 2013. "A Collaborative Framework between a Scheduling System and a Holonic Manufacturing Execution System." In *Service Orientation in Holonic and Multi Agent Manufacturing and Robotics*, edited by T. Borangiu, T. Andre, and D. Trentesaux. 3–17. Berlin Heidelberg: Springer.

Pach, C., A. Bekrar, N. Zbib, Y. Sallez, and D. Trentesaux. 2012. "An Effective Potential Field Approach to FMS Holonic Heterarchical Control." *Control Engineering Practice* 20 (12): 1293–1309.

Pach, C., T. Berger, T. Bonte, and D. Trentesaux. 2014. "ORCA-FMS: A Dynamic Architecture for the Optimized and Reactive Control of Flexible Manufacturing Scheduling." *Computers in Industry* 65 (4): 706–720.

Pinedo, M. L. 2012. *Scheduling: Theory, Algorithms, and Systems*. Berlin: Springer Science & Business Media.

Raileanu, S., M. Parlea, T. Borangiu, and O. Stocklosa. 2012. "A JADE Environment for Product Driven Automation of Holonic Manufacturing." In *Service Orientation in Holonic and Multi-agent Manufacturing Control*, edited by T. Borangiu, T. Andre, and D. Trentesaux. Vol. 402, 265–277. Berlin Heidelberg: Springer.

Russel, S. J., and P. Norvig. 2003. *Artificial Intelligence: A Modern Approach*. 2nd ed. New Jersey: Prentice Hall.

Trentesaux, D. 2009. "Distributed Control of Production Systems." *Engineering Applications of Artificial Intelligence* 22 (7): 971–978.

Trentesaux, D., C. Pach, A. Bekrar, Y. Sallez, T. Berger, T. Bonte, P. Leitao, and J. Barbosa. 2013. "Benchmarking Flexible Job-Shop Scheduling and Control Systems." *Control Engineering Practice* 21 (9): 1204–1225.

Valckenaers, P., H. Van Brussel, P. Verstraete, and B. Saint Germain. 2007. "Schedule Execution in Autonomic Manufacturing Execution Systems." *Journal of Manufacturing Systems* 26 (2): 75–84.

Wilensky, U. 1999. *NetLogo*. Evanston, IL: Center for Connected Learning and Computer- Based Modeling, Northwestern University. http://ccl.northwestern.edu/netlogo/

IBM ILOG CPLEX Optimization Studio [Online]. 2015. Accessed on 2 August 2015. http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud

Yang, T., J. Ma, Z. G. Hou, G. Peng, and M. Tan. 2008. "A Multi-agent Architecture Based Cooperation and Intelligent Decision Making Method for Multirobot Systems." In *Neural Information Processing*, edited by M. Ishikawa, K. Doya, H. Miyamoto, T. Yamakawa. 376–385, January. Berlin Heidelberg: Springer.

Zambrano, G., C. Pach, N. Aissani, T. Berger, and D. Trentesaux. 2011. "An Approach for Temporal Myopia Reduction in Heterarchical Control Architectures". In *Industrial Electronics (ISIE), 2011 IEEE International Symposium on IEEE*. June, 1767–1772.