



**HAL**  
open science

# Dynamic and hybrid architecture for the optimal reconfiguration of control systems: application to manufacturing control

Jose-Fernando Jiménez

► **To cite this version:**

Jose-Fernando Jiménez. Dynamic and hybrid architecture for the optimal reconfiguration of control systems: application to manufacturing control. Artificial Intelligence [cs.AI]. Université de Valenciennes et du Hainaut-Cambresis, 2017. English. NNT : 2017VALE0031 . tel-01668578

**HAL Id: tel-01668578**

**<https://theses.hal.science/tel-01668578>**

Submitted on 20 Dec 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**Thesis submitted for the degree of Doctor from Université de  
VALENCIENNES ET DU HAINAUT-CAMBRESIS**

**Speciality in Automation and Systems engineering**

**Submitted and Defended by Jose Fernando, JIMENEZ GORDILLO.**

**November 7<sup>th</sup> 2017, Valenciennes, FRANCE**

**Doctoral School:**

Ecole doctoral Sciences Pour l'Ingénieur (SPI) – Lille Nord-de-France

**Laboratory and Research department:**

LAMIH Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaines  
Department: Automation and control. Team: SIC – Cooperating intelligent systems.

**DYNAMIC AND HYBRID ARCHITECTURE FOR THE  
OPTIMAL RECONFIGURATION OF CONTROL SYSTEMS.  
Application to manufacturing control**

**JURY**

**REVIEWERS**

GLEIZES, Marie Pierre. Professor at Université Paul Sabatier de Toulouse, Toulouse, FRANCE.  
CARDIN, Olivier. Maître de Conférences HDR at LS2N - IUT de Nantes, Université de Nantes, FRANCE.

**EXAMINERS**

PIERREVAL, Henri. Professor at SIGMA-Clermont, Clermont-Ferrand, FRANCE (President)  
THOMAS, Andre. Professor at Université de Lorraine, Nancy, FRANCE

**DISSERTATION ADVISOR**

TRENTESAUX, Damien. Professor at Université de Valenciennes et du Hainaut Cambresis, FRANCE

**DISSERTATION CO-ADVISORS**

LEITÃO, Paulo. Professor at Instituto Politécnico de Bragança. Bragança, PORTUGAL.  
BEKRAR, Abdelghani. Maître de conférences at Université de Valenciennes et du Hainaut Cambresis.

**Thèse de doctorat pour obtenir le grade de Docteur de l'Université de  
VALENCIENNES ET DU HAINAUT-CAMBRESIS**

**Spécialité automatique et Génie Informatique**

**Présentée et soutenue par Jose Fernando, JIMENEZ GORDILLO.**

**Le 07/11/2017 à Valenciennes, FRANCE**

**Ecole doctorale:**

Ecole doctoral Sciences Pour l'Ingénieur (SPI) – Lille Nord-de-France

**Laboratoire et equipe de recherche :**

LAMIH Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaines  
Department: Automation and control. Team: SIC – Cooperating intelligent systems.

**ARCHITECTURE DYNAMIQUE ET HYBRIDE POUR LA  
RECONFIGURATION OPTIMALE DES SYSTEMES DE  
CONTROLE : Application au contrôle de fabrication**

**JURY**

**RAPPORTEURS**

GLEIZES, Marie Pierre. Professorat Université Paul Sabatier de Toulouse, Toulouse, FRANCE.

CARDIN, Olivier. Maître de Conférences HDR at LS2N - IUT de Nantes, Université de Nantes, FRANCE

**EXAMINATEURS**

PIERREVAL, Henri. Professorat SIGMA-Clermont, Clermont-Ferrand, FRANCE.

THOMAS, Andre. Professor at Université de Lorraine, Nancy, FRANCE

**DIRECTEUR DE THESE**

TRENTESAUX, Damien. Professorat Université de Valenciennes et du Hainaut Cambésis, FRANCE.

**CO-ENCADRANTS**

LEITÃO, Paulo. Professorat Instituto Politécnico de Bragança. Bragança, PORTUGAL.

BEKRAR, Abdelghani. Maître de conférences at Université de Valenciennes et du Hainaut Cambésis.

---

To Catalina, Fernando, Carmen Lucia, Juan Pablo,  
Juliana, Alina, Antonio, Ana Lucia, Carlos, Juan  
Alberto, Daniel, Camila, Julián, Joaquín and  
Clarita

---

## **Jose Fernando Jimenez Gordillo**

Department of Industrial Engineering  
Pontificia Universidad Javeriana  
Calle 40 No 5 – 50, Edif. Jose Gabriel Maldonado S.J.  
Bogotá, Colombia  
[@] j-jimenez@javeriana.edu.co

### **Laboratory Association**

LAMIH, Laboratoire d'Automatique, de Mécanique  
et d'Informatique industrielles et Humaines.  
University of Valenciennes and Hainaut-Cambrésis (UVHC)  
Malvache building  
Le Mont Houy  
F59313 Valenciennes, France.

# Acknowledgements

I would like to express my gratitude to my family, friends, colleagues and lecturers, as well as the organizations that have contributed, at different levels and in different ways, to the completion of this research project. I would like to give some of these special recognition below.

First of all, I would like to express my profound gratitude to my research supervisor, Damien Trentesaux, for his professionalism regarding the development of this PhD work. I acknowledge the trust given to me while conducting this PhD research in a challenging and cutting-edge subject. I recognise that he contributed with both technical and personal skills to the completion of this PhD process. In addition to the valuable teaching at the technical level, specifically concerning distributed control systems, cyber-physical systems and reconfigurable control systems, I would like to emphasize that his encouragement of me to always aim for flawless work have changed my modus operandi in academic, research and consulting activities. I will be always grateful to him for the teaching, advice, discussions and skills learned during my PhD programme.

I would like to thank my co-advisors, Paulo Leitao and Abdelghani Bekrar, for their suggestions, discussions and continuous support in the completion of this research. Prof Leitao provided continuous support for the vision of using research specifically for the applicability and use of the scientific method to improve the quality of this research. Prof Bekrar was a source of support on a daily basis and was instrumental in understanding the collaborative work between the operational research and control communities. I appreciate the availability and interest of both co-advisors during these years.

I would like to thank to Marie Pierre Glaizes, Olivier Cardin, Henri Pierreval and Andre Thomas for agreeing to participate as members of the Jury to review and examine the PhD manuscript and attend the Valenciennes. I express my gratitude for their valuable suggestions to improve this research.

I would like to acknowledge Colciencias and the Pontificia Universidad Javeriana for their support from the beginning of the PhD programme, their unconditional availability for all these years and the financial support they provided me with to live in France during these years. Certainly, this support has allowed me to focus on the development of my research.

I would like to acknowledge the members of the LAMIH Laboratory, the Atelier AIP-PRIMECA Valenciennes and Thérèse Bonte for their generosity and openness in the completion of this research project. Particular gratitude is due to LAMIH, as they enabled me to communicate with outstanding researchers and fellow doctorates that keep me in touch with the latest technological advances in the fields of automatics, mechanics and human control interaction. I express my gratitude for the availability of the AIP PRIMECA at the University of Valenciennes, as this Atelier was established to test and pilot, in a controlled space, the ideas and advances developed from the research. I must also express my enormous gratitude to Mme Bonte, as she helped me get involved in agent-based software, developed the manufacturing simulation as the starting point of my research, and contributed to the improvement of my French language skills over these many years.

## Acknowledgements

---

I would like to specially acknowledge Luis David Prieto, Jorge Sanchez, Jairo Montoya, Karen Gonzalez and Eliana Gonzalez for their support of my starting the PhD programme, and also for their support and encouragement in meeting this challenging goal. Each of them deserves the highest possible gratitude.

Finally, this dissertation and the years spent on it were made possible by many friends and colleagues from my stay in Europe. I want to express my gratitude to Jose Barbosa, Zakaria Yahouni, Cyrille Pach, Gabriel Zambrano, Farid Kadri, Corinne Aureggi, Marlene Genevieve, Sondes Chabbane, Ahmed Hamieh, Wassim Bouazza, Sandro Pires, Jocelene Krevoruczka, Pipocca Pires, Ali Rahimi, Shadab Shishegar, Taisa Goncalves, Salah Eddine Hebaz, Nadya Chouchani, Fadoua Chakchouk, Aminzihno Bekkar, Ahlem Assila and the many researchers that I have met during these years.

# Foreword

The present dissertation is the culmination of doctoral studies at the University of Valenciennes and Hainaut-Cambresis, as part of the project entitled “*SHFMS-II Intelligent Optimization of Centralized/Decentralized Manufacturing and Supply Chain Systems subject to Perturbations*”. This dissertation represents a collaboration by the University of Valenciennes (France), the Polytechnic Institute of Bragança (Portugal) and the Pontificia Universidad Javeriana (Colombia). The completion of this doctoral study and the implementation of this research have been made possible by a conjunction of several professional and personal processes that I would like to describe briefly below.

In 2008, I became interested in an education and research career in industrial and manufacturing engineering after teaching some courses on operational research at Pontificia Universidad Javeriana. During the lectures taught at the industrial engineering department, and after working in parallel in the outbound logistics department at Hewlett Packard for almost five years, I remarked particularly on the lack of innovation and science application in certain Colombian industries. For global companies, the Colombia market is mainly considered a distributor and retailer of products rather than a manufacturer or an innovation developer. A consequence of this situation is that students are trained for the application of foreign models and methodologies that may not entirely apply to the country in question. From my point of view, it is fundamental for Colombia to integrate mechanisms to encourage research and development practices to encourage the industrial innovation process. To this end, as an Assistant Professor at the Javeriana University, I am particularly interested in involving students more actively in the learning process. Therefore, inspired by Piaget’s constructivist view of education and Eric-Mazur’s techniques in interactive teaching, I started encouraging students to construct their knowledge from their own experiences, knowledge, social interaction and personal ideas. However, I also noted that achieving this objective requires further preparation and knowledge in both engineering and research itself. At this point, I realize that a PhD would present an opportunity to start driving research in engineering and education, and to become an active player in the improvement of Colombia’s industry.

For the last decade, Colombia has experienced several positive economic and social changes. After being considered a failed state in the 1980s, Colombia is now a new emerging market beginning to show some of the characteristics of developed countries. For this reason, the government of Colombia and the Colombian Department of Science, Technology and Innovation (Colciencias) have launched a program to support Colombian citizens in undertaking high-level education at the world’s best universities. In 2013, I was awarded this support to pursue PhD studies in Automation and Systems Engineering at the University of Valenciennes. I choose to focus on this domain as I believe that it is the platform of technological advances that offers the best opportunity to develop new solutions and resolve many of society’s problems. It provides systematic and organized leverage to develop important transferable skills to stimulate development in both the educational and industrial contexts. In fact, improvement of competitiveness in science and technology development is the main motivation for undertaking this PhD.

During these years in Valenciennes, I have learned much about the dynamics of research in Europe. It has become clear to me that research activity in France is a highly dynamic environment

that works diligently to implement the results of research. An example of this model, the CIFRE programme, it is a great mechanism that involves developing close relations between researchers, universities and practitioners. I have come to understand that this is definitely the right path to encourage the application and innovation of new solutions, and it is valuable for accelerating the research and development of new technology. Therefore, I am truly satisfied that I could fulfil the expectations that motivates me to undertake the PhD. In addition, I am pleased that during this process I have had many opportunities to interact with various researchers from over the world. I have had several research meetings and stays with other laboratories in France and Europe that have expanded my view of research activities. Without a doubt, I am excited to return to Colombia and benefit from the experience and knowledge acquired during my studies in Europe. Firstly, I am certain that the research I have conducted is a starting point for transforming and improving the competitiveness of Colombian industries. Secondly, I also have no doubt that the experience I have gained will contribute to improvement of skills for achieving interactive teaching objectives.

Jose Fernando Jimenez  
Assistant Professor  
Pontificia Universidad Javeriana

# Extended abstract

## **Dynamic and hybrid architecture for the optimal reconfiguration of control systems: Application to manufacturing control**

Modern society faces the challenge of achieving efficiency, effectiveness and reactivity under exigent demands. Discrete-event control systems, which are among the advances of the information and communication revolution, are emergent technologies that have the opportunity to resolve significant challenges of modern society. In particular, these represent a fundamental solution to manage the new technological advances such that they conform to the increased consciousness of sustainable development. The parameterization, configuration and decision-making of these control systems are critical aspects that impact the performance and productivity required. Dynamic control architecture approaches, such as holonic approaches, dynamic semi-heterarchical architectures and reconfigurable control systems, have been proposed for modelling such systems. In addition, these approaches are able to change their configuration according to the system requirements (improving performance or responding to perturbations), and offer some important features in terms of reactivity, allowing the system to respond to both low- and high-level control. However, such approaches have failed to address the global performance of the control and reconfiguration process after a perturbation. These tend to change the configuration with a focus on the continuity of execution rather than on the optimisation of the reconfiguration. This indicates the need to understand the range and scope of the reconfiguration process in relation to the control process, and the need for a mechanism that optimises the reconfiguration process in order to fulfil the new efficiency, effectiveness and reactivity requirements.

This dissertation proposes a reference architecture for a reconfigurable control system named Pollux. This approach is designed to manage and adjust optimally and in real time the architecture of a control system, either to guide operational execution or to respond to a system perturbation. Composed with a reconfigurable control architecture and the reconfiguration mechanism specified in a set of models and methodologies, Pollux orchestrates the techniques, protocols and interactions for ensuring a certain level of global performance and minimising the degradation caused by perturbations. The contributions of this dissertation are as follows: a) Proposal of a framework for the optimal configuration of control architectures that state the degree of governance between the global coordination and the local operation layers, b) Construction of a flexible and customizable decisional entity model designed to meet optimality and reactivity requirements, c) Proposal of an operating mode model that characterizes the configuration of the control architecture, identifies the specific properties that distinguish its unique capability in the control solution, enables requesting an estimated result if applied in the control system, and provides a comparison reference between different operating modes, and d) Composition of a model with a highly flexible and customizable reconfigurable control architecture that adjusts the configuration for enhancing the efficiency, effectiveness and reactivity of the controlled system.

As methodologies, this dissertation proposes a three-module framework of the reconfiguration mechanism that supports the monitoring, triggering and reconfiguration process, the implementation

in the control system, as well as the general procedures of the reconfiguration that integrate the optimality-based principles into the reconfiguration process.

A reconfigurable control system applied in the manufacturing domain is proposed and validated in a simulation and a real flexible manufacturing system cell located at the University of Valenciennes, France. To this end, three experimental scenarios were conducted to assess the proposed approach. The first experiment showed the feasibility and diversity that result from having different operating modes in a control architecture. The second presents the implementation of optimality-based reconfiguration processes for a real manufacturing system. The final experiment introduced an application of Pollux for dealing with indicators that measure sustainability parameters in the control execution. This validation showed that Pollux minimizes the degradation experienced after a perturbation because of the optimality-based principles included in the reconfiguration mechanism.

**Keywords:** Reconfigurable Control systems, Control systems, Optimization, Reactivity, Dynamic reconfiguration, Multi Agent systems, MAS, Operating mode, Governance, Discrete-event control systems.

# Résumé étendu

## **Architecture dynamique et hybride pour la configuration optimale des systèmes de contrôle. Application au contrôle de fabrication**

La société moderne est confrontée à la contrainte d'efficacité, d'efficacité et de réactivité dans un marché de plus en plus exigeant. Les systèmes de contrôle d'événements discrets, fruit de la révolution des technologies de l'information et de la communication, sont des technologies émergentes qui offrent l'opportunité de résoudre les défis de la société moderne. Ils représentent une solution fondamentale pour conduire les futures avancées technologiques en respectant le développement durable. Le paramétrage, la configuration et la prise de décision de ces systèmes de contrôle sont des aspects critiques qui influent sur les performances et la productivité. Les approches basées sur des architectures de contrôle dynamique, telles que les approches holoniques, les architectures semi-hétéroarchitecturales dynamiques ou encore les systèmes de contrôle reconfigurables ont été proposées pour modéliser ce type de systèmes. Ils peuvent modifier leur configuration pendant leur fonctionnement selon des besoins du système (afin d'améliorer la performance ou de répondre aux perturbations). Ces approches offrent une introspection sur le niveau de réactivité, car le système peut réagir à la fois dans le contrôle de bas et de haut niveau. Cependant, l'inconvénient de ses approches n'arrivent pas à atteindre une performance globale dans la configuration en répondant aux perturbations. En effet, elles ont tendance à changer la configuration en privilégiant la continuité de l'exécution plutôt que l'optimisation de la reconfiguration. Cela indique la nécessité de comprendre la phase du processus de reconfiguration par rapport au processus de contrôle ainsi que la nécessité d'un mécanisme qui optimise le processus de reconfiguration afin de répondre aux nouvelles exigences en matière d'efficacité, d'efficacité et de réactivité.

Cette thèse propose un système de contrôle reconfigurable, nommé Pollux. Ce dernier se veut être une architecture de référence conçue pour gérer et adapter l'architecture d'un système de contrôle de façon optimale et en temps réel, soit pour guider l'exécution opérationnelle, soit pour répondre aux perturbations du système. La thèse décrit de l'architecture de contrôle et du mécanisme de reconfiguration, laquelle sont spécifiés dans un ensemble de modèles et de méthodologies qui définissent les techniques, les protocoles et les interactions pour assurer certaines performances globales et minimiser la dégradation causée par les perturbations. Comme modèles, la thèse apporte les contributions suivantes : a) Propose un modèle de reconfiguration optimale des architectures de contrôle indiquant un degré de gouvernance entre la coordination globale et opération locale du système, b) Propose un modèle d'entité décisionnel flexible et personnalisable répondant aux exigences d'optimalité et de réactivité, c) Propose un mode de fonctionnement caractérisé par la configuration de l'architecture de contrôle, les propriétés spécifiques de la solution du contrôle, la possibilité de l'estimation du résultat lorsqu'il est appliqué et fournit une référence de comparaison entre les différents modes de fonctionnement, et d) Propose une modèle d'architecture de contrôle reconfigurable flexible et personnalisable qui sert de référence à paramétrer avec des techniques prédictives, réactives et de reconfiguration. Comme méthodologies, la thèse propose un mécanisme de reconfiguration qui prend en charge la surveillance, le lancement, le processus de reconfiguration et la

mise en œuvre dans le système de contrôle ainsi que les procédures générales qui intègrent les principes d'optimalité pour la reconfiguration.

Dans la présente thèse, le système de contrôle reconfigurable est instancié pour le domaine manufacturier et est validé dans une cellule réelle d'un système de fabrication flexible situé à l'Université de Valenciennes, en France. Trois scénarios expérimentaux ont été créés pour évaluer l'approche proposée : dans un premier temps, les scénarios montrent la faisabilité et l'intérêt d'avoir des modes de fonctionnement différents dans une architecture de contrôle. Dans un second temps, ils présentent l'application d'un processus de reconfiguration optimale pour un système de manufacture réel basé sur un mécanisme de reconfiguration en parallèle à son fonctionnement. Et finalement, ils incluent une instance de Pollux qui permet inclure des indicateurs de développement durable pendant le fonctionnement du contrôle. Cette validation a montré que Pollux minimise la dégradation après une perturbation en tenant compte des principes d'optimalité inclus dans le mécanisme de reconfiguration.

**Keywords:** système reconfigurable, systèmes de contrôle, optimisation, réactivité, reconfiguration dynamique, systèmes multi-agents, MAS, mode de fonctionnement, gouvernance, FMS, systèmes de contrôle des événements discrets.

# Resumo expandido

## **Arquitetura dinâmica e híbrida para a reconfiguração otimizada de sistemas de controle: aplicação em controle de manufatura.**

A sociedade moderna enfrenta o desafio de alcançar eficiência, eficácia e reatividade sob demandas exigentes. Os sistemas de controle de eventos discretos, como parte dos avanços da revolução das tecnologias de informação e da comunicação, são tecnologias emergentes que têm a oportunidade de resolver desafios significativos da sociedade moderna. Em particular, estes representam uma solução fundamental para gerir os avanços tecnológicos, em direção ao aumento da consciência sobre o desenvolvimento sustentável. A parametrização, a configuração e a tomada de decisão desses sistemas de controle são aspectos críticos que afetam o desempenho e a produtividade. As abordagens de arquiteturas de controle dinâmico, como abordagens holônicas, arquiteturas dinâmicas semi-heterárquicas ou sistemas de controle reconfiguráveis, foram propostas para a modelagem desses tipos de arquiteturas, pois podem alterar sua configuração, durante a execução, de acordo com os requisitos do sistema (por exemplo, melhorar a desempenho ou a resposta às perturbações). Essas abordagens oferecem algumas informações importantes sobre o nível de reatividade, pois o sistema pode responder tanto em baixo quanto em alto nível de controle. No entanto, tais abordagens não conseguiram abordar o desempenho global do processo de controle e reconfiguração depois de uma perturbação, pois estes tendem a mudar a configuração que é focada na continuidade da execução, ao invés da otimização da reconfiguração. Isso indica a necessidade de entender o alcance do processo de reconfiguração em relação ao processo de controle, e a necessidade de um mecanismo que otimize o processo de reconfiguração para cumprir os novos requisitos de eficiência, efetividade e reatividade.

Esta tese propõe um sistema de controle reconfigurável, chamado Pollux, como uma arquitetura de referência projetada para gerenciar e ajustar a arquitetura de um sistema de controle otimizado em tempo real, seja para guiar a execução operacional ou como uma resposta às perturbações do sistema controlado. A arquitetura Pollux engloba uma descrição da arquitetura de controle e do mecanismo de reconfiguração, e especifica um conjunto de modelos e metodologias, orquestra técnicas, protocolos e interações para assegurar desempenhos globais e minimizar a degradação causada por perturbações. Como modelos, nós contribuimos com um framework para a configuração ótima das arquiteturas de controle que indicam o grau de governança da coordenação global ou os níveis de operação local; Uma flexibilidade e um modelo de entidade decisional customizável on-line para alcançar os requisitos de otimização e reatividade, uma definição do modo operacional como modelo para caracterizar a configuração da arquitetura de controle que identifica às propriedades específicas e que distinguem sua capacidade única na solução de controle, fornecendo informações sobre o resultado estimado quando aplicado, e dando uma referência para comparação em diferentes modos de operação; um modelo que cria uma alta flexibilidade e customiza a arquitetura de controle reconfigurável, servindo como referência para parametrizar técnicas de predição, reação e reconfiguração com diferentes níveis de reatividade e otimização. Como metodologias, propomos um framework de três módulos do mecanismo de reconfiguração, o qual suporta o monitoramento, o desencadeamento, o processo de reconfiguração e a implementação do sistema de controle; bem como

os procedimentos gerais da técnica de reconfiguração para integrar os princípios baseados na otimalidade no processo de reconfiguração.

Nesta dissertação, o sistema de controle reconfiguráveis é instanciado no domínio da fabricação e isso é validado por meio de simulações em uma célula real de um sistema de fabricação flexível, localizado em Valenciennes, na França. Para este fim, três cenários experimentais foram conduzidos para avaliar a abordagem proposta. Esses três experimentos mostraram a viabilidade e a diversidade resultantes de se ter diferentes modos de operação em uma arquitetura de controle. Foram incluídos um processo de reconfiguração baseado na otimização, para implementar a melhor alternativa após a reconfiguração, e servem como um estudo de caso para implementar o mecanismo de reconfiguração em paralelo com a execução, para introduzir e instanciar o Pollux, para tratar o desenvolvimento sustentável no controle da execução. Esta validação mostrou que o Pollux minimiza a degradação experimentada após uma perturbação, dado os princípios baseados na otimização, incluídos no mecanismo de reconfiguração.

**Keywords:** Sistemas de controle reconfiguráveis, sistemas de controle, otimização, reatividade, reconfiguração dinâmica, sistemas multi-agentes, MAS, modo de operação, governança, sistemas de controle de eventos discretos.

# Resumen extendido

## **Arquitectura dinámica e híbrida para la reconfiguración óptima de los sistemas de control. Aplicación al control de manufactura.**

La sociedad moderna se enfrenta al reto de lograr eficiencia, eficacia y reactividad bajo demandas altamente exigentes. Los sistemas de control de eventos discretos, como avance de la revolución en la tecnología de información y la comunicación, son tecnologías emergentes que tienen la oportunidad de resolver desafíos significativos de la sociedad moderna. En concreto, representan una solución fundamental para gestionar los avances tecnológicos que buscan una mayor conciencia del desarrollo sostenible. La parametrización, configuración y toma de decisiones de estos sistemas de control son aspectos críticos que impactan el rendimiento y la productividad en los sistemas. En la literatura se han propuesto enfoques de arquitecturas de control dinámico que pueden cambiar su configuración durante su funcionamiento en respuesta a perturbaciones. Enfoques holónicos, arquitecturas semi-heterárquicas o sistemas de control reconfigurables son algunos ejemplos de estas arquitecturas. Las ventajas de estos enfoques es que ofrecen importantes avances en la reactividad, pues el sistema puede responder a varios niveles de control. Sin embargo, estos enfoques no han podido abordar el rendimiento global del proceso de control y reconfiguración en la respuesta de una perturbación. Específicamente, estos sistemas tienden a cambiar la configuración enfocados en la continuidad de la ejecución en lugar de la optimalidad de la reconfiguración. Esto indica la necesidad de entender el alcance de la reconfiguración en relación al proceso de control, y la necesidad de un mecanismo que optimice el proceso de reconfiguración para cumplir con los nuevos requisitos, como los de eficiencia, efectividad y reactividad.

Esta tesis propone una arquitectura de control reconfigurable para eventos discretos, llamado Pollux. Pollux es un sistema de referencia diseñado para gestionar y ajustar la arquitectura de un sistema de control de forma óptima y en tiempo real. Pollux está compuesto por una arquitectura de control y del mecanismo de reconfiguración, que especificado en un conjunto de modelos y metodologías, orquesta las técnicas, los protocolos e las interacciones necesarias para asegurar un rendimiento global y minimizar la degradación causada por perturbaciones. En los modelos, esta tesis contribuye en los siguientes aspectos: a) propone un framework para la configuración óptima de arquitecturas de control que establecen un grado de gobernanza compartido entre los niveles de coordinación global y de operación local de un sistema híbrido; contribuye con un modelo de entidad decisional flexible y customizable diseñados para alcanzar los requisitos de optimización y reactividad requeridos; contribuye con un modelo de *modo de funcionamiento* que caracteriza la configuración de la arquitectura de control, identifica las propiedades específicas que distinguen su capacidad única en la solución de control, y proporciona una referencia de comparación dentro de diferentes modos de funcionamiento; y finalmente, construye un modelo de arquitectura de control reconfigurable, altamente flexible y personalizable, que sirve como referencia para parametrizar con técnicas predictivas, reactivas y de reconfiguración. En las metodologías, la tesis propone un framework de tres módulos para el mecanismo de reconfiguración, que apoyan el monitoreo, detección, arranque, proceso de reconfiguración y la implementación al sistema, así como los procedimientos generales de reconfiguración basados en los principios de optimalidad.

En esta tesis, el sistema de control reconfigurables desarrollado es aplicado en el dominio de la manufactura. Este es validado en una simulación y una célula real de un sistema de manufactura flexible ubicado en la Universidad de Valenciennes, Francia. Para ello, se realizaron tres escenarios experimentales para evaluar el sistema propuesto. En primera instancia, estos experimentos mostraron la factibilidad y diversidad de tener diferentes modos de funcionamiento en una arquitectura de control. En segunda instancia, éstos demostraron la aplicación de un proceso de reconfiguración basado en optimalidad. En tercera instancia, éstos sirvieron como un caso de estudio para implementar el mecanismo de reconfiguración en paralelo a la ejecución. Y, Finalmente, éstos introdujeron una instancia de Pollux para tratar indicadores de sostenibilidad durante el funcionamiento. Esta validación mostró que Pollux minimiza la degradación experimentada después de una perturbación dado los la inclusión de principios de optimalidad incluidos en el mecanismo de reconfiguración.

**Keywords:** Sistemas de control reconfigurable, sistemas de control, optimización, reactividad, reconfiguración dinámica, sistemas multi-agente, MAS, modo de operación, gobernabilidad, sistemas de control de eventos discretos.

# List of Acronyms

<b>ADACOR</b>	Adaptive holonic control architecture for distributed manufacturing systems
<b>ADACOR<sup>2</sup></b>	Evolved version of ADACOR
<b>AGV</b>	Automatic guided vehicle
<b>AI</b>	Artificial intelligence
<b>AIP-PRIMECA o AIP</b>	Atelier Inter-établissement de Productique (Pôle de Ressources Informatique pour la Mécanique).
<b>BDA</b>	Big data analytics
<b>CA</b>	Control architecture
<b>CPS</b>	Cyber-Physical systems
<b>D-CA</b>	Dynamic control architecture
<b>D-MAS</b>	Delegated multi agent systems
<b>DES</b>	Delivery systems
<b>ERP</b>	Enterprise resource planning
<b>FJSP</b>	Flexible job shop problem
<b>FMS</b>	Flexible manufacturing system
<b>FPGA</b>	Field-programmable gate arrays
<b>GA</b>	Genetic algorithm
<b>GDE</b>	Global decisional entities
<b>ICT</b>	Information and communication technology
<b>IoT</b>	Internet of things
<b>IT</b>	Information technology
<b>LDE</b>	Local decisional entities
<b>LPT</b>	Longest processing time
<b>MES</b>	Manufacturing execution system
<b>MILP</b>	Mixed integer linear programming
<b>MO-GA</b>	Multi-objective genetic algorithm
<b>ORCA</b>	Optimized and reactive control architecture

## List of Acronyms

---

<b>PROSA</b>	Product Resource Order Staff Architecture
<b>RCAs</b>	Reconfigurable control architectures
<b>RCS</b>	Reconfigurable control system
<b>RDE</b>	Resource decisional entities
<b>RM</b>	Reconfiguration mechanism
<b>RT</b>	Reconfiguration time
<b>Rtrig</b>	Reconfiguration triggering
<b>Rtech</b>	Reconfiguration technique
<b>Rsync</b>	Reconfiguration synchronization
<b>S-CA</b>	Static control architectures
<b>SCADA</b>	Supervisory control and data acquisition
<b>sGDE</b>	Sustainable global decision entity
<b>sLDE</b>	Sustainable local decision entity
<b>SPT</b>	Shortest processing time
<b>sRDE</b>	Sustainable resource decision entity
<b>TDM</b>	Task distribution module
<b>UN</b>	United Nations
<b>WLAN</b>	Wireless local area network

# Table of Contents

<b>Introduction.....</b>	<b>1</b>
1. Background.....	1
2. Research question .....	3
3. Objective and contribution.....	4
4. Dissertation structure .....	4
<b>Chapter 1 .....</b>	<b>6</b>
<b>Control architectures: Towards a reconfigurable architecture .....</b>	<b>6</b>
1.1 Introduction.....	6
1.2 Modern society.....	6
1.2.1 The information and communication technologies revolution .....	7
1.2.2 Societal Development .....	9
1.2.3 ICT: a contribution to solve the societal needs .....	10
1.3 Theory and control for discrete-event systems .....	12
1.3.1 System theory.....	12
1.3.2 Control systems.....	13
1.4 Configuration of control architectures .....	19
1.4.1. From static configuration of control architectures.....	19
1.4.2. ...to dynamic configuration of control architectures .....	21
1.5 Summary .....	22
<b>Chapter 2 .....</b>	<b>23</b>
<b>Reconfigurable control systems: literature review .....</b>	<b>23</b>
2.1 Introduction.....	23
2.2 Theoretical framework of reconfigurable control systems .....	23
2.2.1 Reconfigurable control systems: technological solution.....	24
2.2.2 Purpose of reconfigurable control systems .....	27
2.2.3 Means of reconfigurable control systems .....	28
2.3 Literature review .....	32
2.3.1 Methodology of the literature review.....	33
2.3.2 Characterization and analysis of reconfigurable control systems .....	38
2.4. Summary .....	44
<b>Chapter 3 .....</b>	<b>46</b>

<b>Pollux: a reconfigurable control system .....</b>	<b>46</b>
3.1 Introduction.....	46
3.2 Specifications of the Pollux approach.....	46
3.3 Pollux: a reconfigurable control system .....	49
3.3.1 Description of the proposed approach.....	49
3.3.2 Control architecture.....	49
3.3.3 Reconfiguration mechanism.....	59
3.4 Reconfiguration technique for improving optimality and controllability.....	64
3.5 Summary .....	66
<b>Chapter 4 .....</b>	<b>68</b>
<b>Application to the control of flexible manufacturing systems .....</b>	<b>68</b>
4.1 Introduction.....	68
4.2 Overview of flexible manufacturing systems .....	68
4.3 Description of the FMS control problem.....	69
4.3.1 Formulation of the flexible job shop problem.....	69
4.3.2 Decision making in the Flexible job shop problem .....	71
4.3.3 Performance indicators for the flexible job shop problem.....	72
4.4 Pollux applied to flexible manufacturing systems.....	75
4.4.1 External view .....	75
4.4.2 Internal view.....	77
4.2.3 Reconfiguration mechanism.....	83
4.5 Summary .....	88
<b>Chapter 5 .....</b>	<b>90</b>
<b>POLLUX-AIP: An experimental case study .....</b>	<b>90</b>
5.1 Introduction.....	90
5.2 Valenciennes AIP-PRIMECA: a flexible manufacturing system.....	90
5.2.1 Description of the manufacturing cell.....	91
5.2.2 Mathematical formulation of the Flexible manufacturing cell .....	95
5.3 Pollux implementation to the AIP cell (Pollux-AIP).....	97
5.3.1 External view: Purpose and environment .....	98
5.3.2 Internal view: control architecture and reconfigurable mechanism.....	98
5.4 Experimental protocol.....	99
5.4.1 Experiment A. ....	99
5.4.2 Experiment B. ....	106
5.4.3 Experiment C. ....	114
<b>Conclusions and future work.....</b>	<b>122</b>
<b>References.....</b>	<b>126</b>

<b>APPENDICES .....</b>	<b>140</b>
<b>Appendix A .....</b>	<b>141</b>
<b>Appendix B .....</b>	<b>144</b>
<b>Appendix C .....</b>	<b>146</b>
<b>Appendix D .....</b>	<b>152</b>
<b>Appendix E .....</b>	<b>158</b>

# List of Figures

<b>Figure 1</b> Fulfilment of societal needs through technological advances .....	11
<b>Figure 2</b> Nature of a control system and its characteristics (adapted from Trentesaux (2002)). .....	14
<b>Figure 3</b> Model of the environment in a control system (Trentesaux 2002).....	15
<b>Figure 4</b> Internal view of the control system and composition of a decisional entity .....	16
<b>Figure 5</b> Types of control system structures (Adapted from Dilts, Boyd, and Whorms (1991)) .....	17
<b>Figure 6</b> Reconfigurable control systems within the context of ICT as a solution to meet societal needs. ....	24
<b>Figure 7</b> General framework of the reconfigurable control system .....	28
<b>Figure 8</b> Illustration of the Reconfiguration processes in reconfigurable control architecture. ....	30
<b>Figure 9</b> Evaluation attributes of the reviewed reconfigurable control systems.....	34
<b>Figure 10</b> Static, composition-related features of reconfigurable control systems.....	35
<b>Figure 11</b> Dynamic features of reconfigurable control systems .....	38
<b>Figure 12</b> Changing the configuration of the control architecture to attain optimal performance ....	47
<b>Figure 13</b> General illustration of the reconfigurable control architecture of Pollux.....	49
<b>Figure 14</b> Composition and functioning of the decisional entity.....	50
<b>Figure 15</b> Communication protocol between decisional entities. ....	53
<b>Figure 16</b> Mapping of the possible configuration of the Pollux control architecture .....	54
<b>Figure 17</b> Relation between the governance parameters and the operating mode. a) Example of control architecture, b) illustration of the representation of the operating mode through a structured array, and c) illustration of the range of operating modes .....	56
<b>Figure 18</b> Dynamic process of the control architecture .....	58
<b>Figure 19</b> General process of the reconfiguration mechanism and its control over the control architecture.....	59
<b>Figure 20</b> General process of the reconfiguration mechanism .....	60
<b>Figure 21</b> Sequence diagram of Pollux, including the preparation phase, task allocation phase and execution phase .....	62
<b>Figure 22</b> Process of the reconfiguration control technique to obtain the optimal configuration .....	64
<b>Figure 23</b> General process of the reconfiguration technique .....	65
<b>Figure 24</b> Example of a Flexible job shop system .....	71
<b>Figure 25</b> Architecture of Pollux applied to the manufacturing system. ....	75
<b>Figure 26</b> Location of the proposed approach within an industrial enterprise system (External view) .....	76
<b>Figure 27</b> Structure of Pollux for the FJSP (Internal view).....	77

<b>Figure 28</b>	Illustration of global decisional entity GDE a) composition, and b) functional diagram .	79
<b>Figure 29</b>	Illustration of local decisional entity LDE a) composition, and b) functional diagram ...	81
<b>Figure 30</b>	Illustration of resource decisional entity RDE. a) composition, and b) functional diagram .....	82
<b>Figure 31</b>	Reconfiguration mechanism functioning over the control system .....	83
<b>Figure 32</b>	Functional diagram of the reconfiguration mechanism .....	85
<b>Figure 33</b>	Conceptual model of relation of the progress of completion and the time of execution (Inspired from Rojas-Ramirez (1999)). .....	86
<b>Figure 34</b>	Comparison of Pollux with other approaches in an example for the scheduling of 21 jobs into 4 resources. ....	87
<b>Figure 35</b>	AIP-PRIMECA: Flexible manufacturing system located in University of Valenciennes .....	91
<b>Figure 36</b>	Layout of the AIP-PRIMECA flexible manufacturing cell, Valenciennes. ....	93
<b>Figure 37</b>	Types of job processed by the AIP cell .....	94
<b>Figure 38</b>	<b>Components of the AIP cell</b> .....	94
<b>Figure 39</b>	Information of the AIP cell, a) Production sequence per jobs, b) processing time component, and c) transport time between machines. ....	95
<b>Figure 40</b>	Pollux implementation to the AIP Primeca, a) relation virtual-physical system, and b) Pollux architecture for FMS.....	97
<b>Figure 41</b>	Implementation of Pollux for experiment A.....	100
<b>Figure 42</b>	Results of the simulation of different operating modes in the proposed model .....	104
<b>Figure 43</b>	a) Makespan comparison for different control systems (b) Results of the execution of scenario .....	105
<b>Figure 44</b>	Implementation of Pollux for experiment B.....	107
<b>Figure 45</b>	Sequence diagram of the reconfiguration mechanism for experiment B .....	110
<b>Figure 46</b>	Description of the functioning of the reconfiguration mechanism in the production execution .....	111
<b>Figure 47</b>	Results of the GA-Tuning.....	112
<b>Figure 48</b>	Makespan comparison for three different execution strategies .....	113
<b>Figure 49</b>	Implementation of Pollux for experiment C.....	115
<b>Figure 50</b>	Go-green holon in manufacturing (Trentesaux and Giret 2015) .....	116
<b>Figure 51</b>	Schematic view of the experimental case.....	118
<b>Figure 52</b>	Simulations of scenario 4 for Approach B & C.....	120

## List of Tables

<b>Table 1</b>	Reconfiguration approaches in reconfigurable control systems .....	27
<b>Table 2</b>	Characterization of selected papers from the literature review of reconfiguration control systems.....	39
<b>Table 3</b>	Typology of the reviewed literature according to the reconfiguration technique.....	43
<b>Table 4</b>	Characterization of Pollux regarding the static and dynamic features of the architecture....	67
<b>Table 5</b>	Static and dynamic indicators for the manufacturing domain. ....	74
<b>Table 6</b>	Level for each factor of experiment.....	102
<b>Table 7</b>	Description of the proposed control systems .....	103
<b>Table 8</b>	Detailed description of decisional entities responsibility and interactions .....	108
<b>Table 9</b>	Parameters for the genetic algorithms.....	110
<b>Table 10</b>	Comparison of approaches after the perturbed scenario .....	119

# Introduction

## 1. Background

Systems face the challenge of achieving efficiency, effectiveness and reactivity under exigent demands. To meet this challenge, control systems are implemented in several domains, including engineering, aeronautics, robotics, health care, and many other systems that aim to achieve the same objectives. The purpose is to improve operational performance and increase the reactivity for featuring a productive and continuous operation. In practice, control systems represent a fundamental solution to manage designated operations for goal-oriented systems.

In general terms, a **control system** is defined as a system that manages the behaviour of other systems. There are different types of control systems depending on the events in these systems, the functioning and the field of application. A control system may be a continuous-event or discrete-event system based on the type of variables used, or an open-loop or closed-loop system depending on whether or not it contains feedback. Control systems can be also classified according to the field of application, which may include mechanical, electronic, pneumatic, computer, or even financial or commercial systems. For example, while a control system for a washing machine is a closed-loop and continuous control system designed for an electrical system, an automated manufacturing plant is closed-loop and discrete-event control system designed for a hybrid system (e.g. mechanical, electrical, pneumatic, among others). For the sake of this study, this dissertation focusses on closed-loop control systems for a discrete-event system.

The parameterization, configuration and decision-making of control systems are critical aspects that impact the performance, productivity and reactivity required. Originally (and still in some cases), control systems were built under centralized architectures (also known as hierarchical architectures), whose foremost objective is to ensure efficient operation regardless of the need for reactivity in unexpected scenarios. In response to this drawback, researchers have explored decentralized architectures (also known as heterarchical architectures) that feature reactivity capabilities in unexpected or perturbed scenarios. Indeed, although the reactivity requirement is improved with this approach, it does not achieve the same level of efficiency as centralized architectures. Under these circumstances, an approach is required that features the benefits of both hierarchical and heterarchical approaches in order to contribute to the performance, productivity and reactivity required.

Recently, researchers have shown an increased interest in approaches that improve the global performance and reactivity goals from hierarchical and heterarchical architectures. In general, these approaches encourage the inclusion of a flexible configurations to orchestrate improved functioning of the control architecture. However, combinatorial possibilities remain to articulate this architecture. In fact, these configurations may also change with time, as trends may evolve towards a more hierarchical

architecture, a more heterarchical architecture or a combined architecture, for example. Concerning this characteristic, there are two categories: static and dynamic configuration of the control system. While static approaches, such as semi-heterarchical architectures and predictive-reactive approaches, maintain their configuration regardless of the events during execution, dynamic approaches, such as holonic systems and reconfigurable control systems, can change their configuration during execution according to the system requirements. Although both categories represent efforts to achieve an efficient combination of hierarchical and heterarchical architectures, the difference between these two categories is the level of reactivity featured in respect to perturbations.

Within the dynamic approaches, the changes of control architecture may be executed according to different perspectives either to recover or improve global performance. These perspectives are: a) *change of parameters*, which refers to control systems that modify either a global parameter of the system or a component specification to influence the behaviour of the control system (e.g. re-tuning, change of process, change of product); b) *change of configuration*, which refers to control systems that modify the control architecture in terms of structural arrangement, behaviour function and dynamism progression (e.g. reconfigurable control system, evolving systems, holonic systems); and c) *changes in the decision variables*, which refer to control systems that modify the decisions and planned actions as responses to perturbation (e.g. rescheduling, repairing, robust approaches). This dissertation focusses on the **change of configuration approach**, specifically in reconfigurable control systems.

The use of reconfigurable control systems is a promising solution that enhances system performance, improves productivity and supports the reactivity required in control systems. Reconfigurable control systems are capable of changing the characteristics of the control architecture automatically and in real time to accommodate large variations in the process being controlled, thus maintaining the stability and performance of the original system as much as possible (Jiang 1994). This approach addresses the stated challenge, as it monitors the system and configures a suitable configuration according to the system needs. Despite being flexible enough to feature changeable configurations and customizable to system conditions, there remain some challenges to resolve in order to fully exploit the benefits of such control systems. First, the possible alternatives that can be adopted as a result of a reconfiguration process are limited to a few configuration options, mostly from pre-determined or loosely pre-designed and pre-evaluated possibilities. Second, the reconfiguration process focusses mainly on ensuring the continuity of system execution regardless of the efficiency achieved during this process. The functioning of the reconfigurable control system without facing these challenges promises a reactive execution that recovers stability and equilibrium in the event of unexpected perturbations. Nevertheless, this process does not guarantee an optimal recovery to the same expected performance, or even minimize the degradation caused after the perturbation. This deficiency of optimality recovery comes about mainly because the reconfigurable process lacks controllability tools that steer the system to a desired state. In sum, reconfigurable control systems need to feature an optimal and controllable process in order to truly meet efficiency, effectiveness and reactivity requirements.

Optimization techniques used in operational research may contribute to the improvement of the reconfiguration process. These techniques use iterative methods and algorithms that seek out the best option from a set of alternatives in terms of a specific metric or objective function. These widely applied techniques have the potential to provide optimal or adequate solutions to large and complex problems. However, a disadvantage of these techniques is that they require a considerable amount of computational time to reach this solution. Nevertheless, the framework and the principles used in the

optimization may provide an inspiration to design and develop a framework for the reconfiguration process.

## 2. Research question

Considering the aforementioned background and the fact that implementation of reconfigurable control systems represents a clear opportunity to achieve the efficient, effectiveness and reactive systems required in modern society, the main research question of this doctoral dissertation is: **How can the reconfiguration process be improved in order to enhance the global performance achieved by reconfigurable control systems?** This question raises three specific sub-questions:

- 1) Which factors from the control architecture and the reconfiguration process impact the global performance of the reconfigurable control system (RCS)?
- 2) Can global performance enhancement of the RCS be achieved by including optimality-based principles in the reconfigurable process?
- 3) Considering the global performance of the system, to what extent can a recurrent change of configuration sought out with optimality-based principles mitigate the degradation caused by perturbations?

The position adopted in this doctoral dissertation can be summarized as follows:

*The inclusion of optimization-based principles in the reconfiguration process can improve the controllability and optimality characteristics of reconfigurable control systems because a guided search process makes possible to explore the space of alternative organizations of the control architecture and find an adequate configuration such that its implementation recovers the expected global performance or at least minimizes the degradation from perturbations.*

This statement is supported by development of the following research pillars:

- A reconfigurable control system, which is a dynamic architecture coupled between hierarchical and heterarchical architectures, benefits simultaneously from the advantages of these approaches in terms of obtaining efficient performances, improves the productivity of the system and supports the reactivity of its execution under both normal and perturbed conditions.
- The components of the control system, whose constitution can be flexible, customizable, modular, convertible and diagnosable, can be built with a decision-based approach, containing diverse decisional processes from advanced and complex decision-making techniques to render prompt decisions from condition-action rules.
- The diverse combinations of the coupled architecture can be used as different control strategies and each characterization represents a configuration that steers the control system to a desired system state.
- Optimality-based principles can be used for the reconfiguration process as they can provide an adequate configuration of the coupled architecture and consequently contribute to the optimality and controllability of the reconfigurable control system.

### 3. Objective and contribution

This dissertation addresses the stated challenge and presents a reconfigurable control system intended specifically for the control of discrete-event systems. This reconfigurable control system, named Pollux, is a reference architecture designed to manage and adjust optimally and in real-time the architecture of a control system, either to guide operational execution or to respond to a system perturbation. The objective of Pollux is to improve the performance of the reconfiguration process of a reconfigurable control system through a specific change in the characteristics of the control architecture in terms of the structural arrangement, behavioural functioning and dynamism of the control system.

The reader should bear in mind that the study is based on the orchestration of multiple technologies (i.e. mathematical programming, potential fields, multi-agent systems, and others) to provide the recovery and stability required by the control system. Due to practical constraints, this dissertation does not address these technologies as an innovation, since this study is intended to use these technologies rather to conduct new theoretical research in this area. In addition, other perspectives for the reconfiguration of the control system (e.g. rescheduling, repairing, etc.) and the synchronization of these perspectives with the reconfiguration of the control architecture are out of the scope in this dissertation. Finally, given that the reconfiguration process encompasses the monitoring, reconfiguration and synchronization of the new configuration, this dissertation does not perform a compressive examination of monitoring and synchronization, and instead focusses on the reconfiguration process rather than the detection and implementation part of the reconfiguration.

The thesis of this dissertation and the developed reconfigurable control systems are validated in a simulation and the real cell of a flexible manufacturing system located in Valenciennes, France. To this end, three experimental scenarios were conducted to assess the proposed approach, where similar approaches without reconfiguration capabilities are compared.

### 4. Dissertation structure

The structure of this dissertation is as follows:

**Chapter 1**, entitled “*Control architectures: Towards a reconfigurable architecture*”, contextualizes this dissertation around the configuration of the architectures of control systems (either static or dynamic configurations) as it can be used as framework to manage and orchestrate the use of technological advances to resolve some of the current global challenges. This chapter explains important concepts in the external and internal view of the control system, characterizes the static and dynamic configuration of control architectures, and highlights the knowledge gap concerning the dynamic configuration of control architectures.

**Chapter 2**, entitled “*Reconfigurable control systems: literature review*”, reviews the literature concerning reconfigurable control systems that feature an optimality initiative in the reconfiguration process. The chapter discusses the contribution of the approaches reviewed and characterizes these according to the composition-related (e.g. structure arrangement and behaviour functioning) and reconfiguration-related features (e.g. representation of control solution, purpose, process, quality and

optimality level of the reconfiguration) of the reconfigurable control system. Finally, the chapter categorizes these based on the theoretical framework and describes the relationship of these contributions to the reconfiguration characteristics.

**Chapter 3**, entitled “*Pollux: a reconfigurable control architecture*”, provides a detailed description of Pollux. The general objective of Pollux is feature a reconfiguration mechanism that steer the control architecture towards recovery and stability in the event of a perturbation, while it simultaneously improves the efficiency, effectiveness and reactivity of the control system. First, the concept of changing of the control architecture to attain the optimal configuration is described, as this is the basis of the proposed approach. The general specifications of Pollux are then described, taking into consideration the design of the reconfiguration control system. The rest of the chapter focusses on the description of the proposed approach, including control architecture and the functioning of the reconfigurable mechanism.

**Chapter 4**, entitled “*Application to the control of flexible manufacturing systems*”, presents a case study in which Pollux is implemented in a general Flexible Manufacturing system (FMS). First, an overview of a FMS is presented, including the flexible job shop problem (FJSP). Then, the control objectives, system objectives, decision variables, parameters, assumptions and constraints of the FJSP are introduced as they relate to the case study. Finally, the proposed approach, Pollux, is implemented by configuring the external and internal view of the reconfigurable control system, including its components, control architecture and reconfigurable mechanism.

**Chapter 5**, entitled “*POLLUX-AIP: An experimental case study*”, presents the assessment of the proposed approach using as a case study a specific FJSP from an operating FMS located in Valenciennes (France). This chapter describes the application of Pollux to the global control problem of the FJSP and the benefits of Pollux compared to other control solutions for discrete-event systems. For the assessment, three experiments are conducted in order to evaluate the recovery and stability of the system performance when a perturbation occurs. While the first two experiments are targeted on the performance of a single criteria, the third experiment includes a multi-criteria evaluation.

Finally, this dissertation concludes by summarizing the contributions of this research in the “*Conclusion and future work*” section. First, general conclusions are presented regarding the design and implementation of Pollux. The contribution of Pollux to the reconfiguration process is summarized and recommendations for implementation are presented. The limitations of the current research are then presented along with suggestions of how to overcome these limitations. Last, guidelines for further research in reconfiguration control architectures are suggested and the future research by the author derived from this dissertation is outlined.

# Chapter 1

## Control architectures: Towards a reconfigurable architecture

### 1.1 Introduction

The objective of this chapter is to present the context of this dissertation. Section 2 highlights the opportunity of using technological advances to resolve significant challenges. To this end, it describes some emergent fields from the current technological revolution, presents the expectations demanded by society, and finally reveals the underlying structure of the approach to technological advances as a solution to these expectations. Considering the need to develop control systems able to support the required needs, certain concepts related to control systems are presented in Section 3. A description of the configuration of control architectures and the characterization of configuration classes are presented in Section 4. It highlights a knowledge gap in the dynamic configuration of control architectures in the domain of the control system. Finally, a chapter summary is presented in Section 5.

### 1.2 Modern society

The social, economic and environmental interests of modern society have change over the last 30 years. The development of technology, the globalization of markets, the decentralization of operations, and the increased consciousness of sustainable development have compelled modern society to rethink the strategy and techniques in the management of various systems, including manufacturing, logistics, operations and social systems (Gunasekaran & Ngai 2012). The next decade is likely to see a considerable increase in the complexity of systems due to new requirements in terms of efficiency, effectiveness and reactivity. In order to identify and address this complexity, the development of modern society can be conceptualized as an agreement between two evolving forces: the **push forces of information and communication technology (ICT) advances** and the **pull forces of societal needs**.

### 1.2.1 The information and communication technologies revolution

A technological revolution consists of a strong and dynamic cluster of new technologies, products and industries that can lead to growth and long-term development (Radzievska 2016). There are several push forces that encourage the use of new advances on a daily basis, such as ICT, mechanical engineering, natural sciences, and others. In the specific field of ICT, new devices and software applications are bringing about a technological revolution that radically changes not only how we see the world, but will also change how we interact with it (Mohamed, Murray & Mohamed, 2010). One important aspect of this revolution is related to transforming the visualization of the world as a collection of smart communicating objects that interact together to achieve a specific purpose. A smart communicating object can be broadly defined as a physical/digital object which is augmented with sensorial, processual and network capabilities (Kortuem, Kawsar, Sundramoorthy & Fitton, 2010). These objects essentially exploit the use of hardware, software and communicating technologies to enhance the collective capability of humankind. For example, an application of smart communicating objects can be found in agriculture. New technologies monitor the growing conditions in potato fields (i.e. hydrological and thermal stress, soil properties, pH levels, and others) to improve farming methods, protect the environment and increase the sustainability of the agricultural operations (Rad, Hancu, Takacs & Olteanu, 2015). Like the industrial revolution in terms of its impact in the social-cultural context (Kagermann et al., 2013), this revolution is inspiring the use of technology in everyday life that may soon change the standard of living.

Several technologies are being developed to deploy smart communicating objects. Some examples of these technologies are cyber-physical systems, big data analytics, the internet of things, artificial intelligence, automation, web services, cloud computing, service orientation, embedded systems, additive manufacturing, and many others. Each of these disciplines provides key opportunities for the development of this ICT revolution engaged from different perspectives. In fact, these technologies are generating considerable interest among practitioners and researchers, as they have the potential to resolve some of the most difficult challenges across various domains (Lee 2015). A general description of selected technologies is presented below.

**Cyber-Physical systems (CPS)** are recognized as one of the main technologies to design and build effective systems to leverage the opportunities provided by this revolution. A CPS is a collection of communicating computing devices that interact with the physical world via sensors and actuators in a feedback loop (Alur 2015). This provides the infrastructure foundation of future systems and forms the basis of emerging and future smart services (NIST 2017). Although this concept is in the initial stage of deployment, it has several functionalities that assist in the design and development of high-technology innovations (Lee 2015). In brief, a CPS is capable of sensing and providing proactive knowledge of the physical world, and it rapidly processes information for real-time analytics, features capabilities for collaborative and/or autonomous decision-making, connects with other components vertically and across industries, and features high-level control and process automation. CPS is a platform that permits orchestration of the networked computational resources that handle the function of the physical world and create value in a specific industry (Leitão, Colombo & Karnouskos, 2016). Some applications of CPS include unmanned aerial vehicles (Jamshidi, Jaimes Betancourt & Gomez, 2011), smart buildings (Gurgen et al. 2013), precision agriculture (Cimino et al. 2017) and ethical autonomous decisions (Trentesaux, & Rault, 2017a).

**Big data analytics (BDA)** is another technique that makes use of the opportunities inherent in the ICT revolution. The concept of BDA, as an application of advanced analytics to big data sets, refers to the techniques and tools that aim to gain information insights to drive and support decision-making (Russom 2011). BDA is a collection of techniques, processes and methodologies (e.g. predictive analytics, data mining, statistical analysis and database methods) that analyse vast amounts of data to understand the information contained and provide valuable input to the decision-making process. The overall process of BDA can be broken down into the following steps (Gandomi & Haider 2015): 1) acquisition and recording of raw data, 2) extraction of the structured data, 3) pre-processing of data, executing data aggregation and/or representation, 4) execution of analytics models, and 5) the decision-making process itself. Some applications of big data analytics can be found in the oil industry (Hegde & Gray 2017), healthcare (Wang & Hajli 2017) and e-Logistics (Yu et al. 2016).

Most of the opportunities of the ICT revolution will be enhanced using the **internet of things (IoT)**. The IoT is recognized as one of the most important areas of future technology and is gaining much attention from a wide range of industries (Lee & Lee 2015). The IoT paradigm, sometimes called the internet of everything, is based on the fact that everyday objects can be equipped with identifying, sensing, networking and processing capabilities that, through unique addressing schemes, are able to interact and cooperate with each other to accomplish a specific objective or objectives (Whitmore et al. 2015; Atzori et al. 2010). The IoT is seen as an emerging area of technology that represents the backbone of novel high-tech advances and provides the required computational resources for creating revolutionary capabilities (Gubbi et al. 2013). A recent survey about the application, trends and challenges of the internet of things can be found in (Ali Shah et al. 2017).

**Artificial intelligence (AI)** is another constantly evolving technology that shows the opportunities of this ICT revolution. This revolution is considered as a major driver of economic growth and social progress if industry, civil society, governments and the public work together to support the development of such technologies (National Science and Technology Council 2016). AI is a system capable of solving of problems using computer applications by methods modelled after natural activities and human cognitive processes (Schalkoff referenced in (Rutkowski 2008)). The main areas of AI are knowledge-based systems, computer sensory systems, language processing systems, bio-inspired algorithms and machine learning. Therefore, AI provides a virtual representation of the smart objects and intelligent processing capabilities to solve human challenges, and serves as a simulation tool that emulates real-life situations to support decision-making processes. It includes the ability to design intelligent agents that perceive their environments and take actions that maximize their chances of success in meeting an objective (Russell & Norvig 2002). Some applications of artificial intelligence are face detection/recognition (Hjelmås & Low 2001), manufacturing control (Thomas, Trentesaux, Valckenaers, 2012), robotics (Brambilla et al. 2013), machine learning (Domingos 2012), Building information modelling (Hamieh et al. 2017), and optimization through intelligent products (Bouazza et al. 2015).

**Automation**, or automation control, has also contributed to the ICT revolution. Although there are several definitions, automation refers to any technique, method or system that operates and/or controls a process automatically, by means of electronic devices or software applications, and minimizes human intervention (Rada & Holden 2009). Initially, this discipline started in the manufacturing sector because industrial automation offered the ability to respond quickly to change while maintaining stable and efficient operations. Automation is also a key technology in enhancing

industrial competitiveness (Brennan et al., 2008). However, automation significantly improves the performance of processes in any domain. It aims to ease the meeting of specific requirements that would ordinarily be difficult with human skills, such as specifications in speed, precision and frequency, among many others. Some applications other than manufacturing include energy (Gungor & Lambert 2006), and navigation systems (Demesure et al. 2016)

**Web services, service-orientation and cloud computing technologies** are recent ICT developments that have a great potential and will certainly be used in the interconnected world. These technologies consider the use of services as modular and dynamic information to be used on demand through the Internet (Paolucci et al. 2002). In general, such services provide data and processing resources throughout the internet to be used by other technologies, such as service-orientated architectures (Morariu et al. 2016). This kind of technology is defined as a “model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” (Mell & Grance 2011). Some examples with this technology are described in Quintanilla et al., (2016), Mircea & Andreescu (2011), and Hashemi, Monfaredi & Masdari (2013)

Industries and government agencies are aligned to derive the most profitable scenario from the ICT revolution. There are encouraging innovation policies based on the use of technology. Some examples of these policies can be seen in France (DGE – Ministère de l'économie, de l'industrie et numérique, 2016), Germany (Acatech – Deutsche akademie der technikwissenschaften, 2012) and the United States (NIST – U.S. Department of Commerce, 2012). Industries such as manufacturing and energy are challenged, from their own perspective, in achieving their objectives because of daily problems and inefficiencies caused by their inherent limitations. However, the technological push forces are an opportunity to exploit new development in the service of humanity. Certainly, innovation with a technology push strategy does not necessarily meet human needs. Nevertheless, the implementation of these technologies is an accelerated process that orchestrates the work of multidisciplinary domains to exploit new discoveries for human development.

### 1.2.2 Societal Development

Current global challenges are directly related to difficulties of an economic, social and environmental nature (Gunasekaran & Ngai 2012). The complexity of these challenges lies in the fact that social and environmental goals often conflict with economic ones (Montoya-Torres 2015). These challenges have different perspectives depending on the field involved, such as sustainability, security or maintainability, among others. Still, the United Nations (UN) suggests the adoption of a sustainable development model to tackle these challenges. **Sustainable development** can broadly be defined as the expansion or progress of an organization that meets the needs of the present without compromising the ability of future generations to meet their own needs (World commission on Environment and Development, 1987). This encompasses the principle of ensuring that our actions today do not limit the range of options open to future generations (Elkington, 1998). The objective of this framework is to encourage all organizations to set the course of their operations with this in mind, because the world is currently facing devastating challenges such as climate change, income disparity, unemployment, population growth, and many others. Unfortunately, in the mid- to long-term future, these challenges directly affect the *economic prosperity*, *environmental quality* and *social well-being* of future

generations (Hacking & Guthrie 2008). As a consequence, including a sustainable development model increases complexity in organizations and systems (Rosen & Kishawy 2012). A transition to a sustainable society requires extensive coordinated collaboration across disciplines and sectors (Broman 2017). Still, experts agree that there is a need to focus on resolving two main concerns in order to address these challenges properly (World Economic Forum Report in Risks, 2017). The first is how to manage systems whose goals are related to sustainable development objectives? And, second, what is the best way to handle the interconnection between the economic, social and environmental dimensions within this system?

The implementation of a sustainable development model requires breaking down responsibilities at the strategic, tactical and operational level, according to the objectives of the organization (Stadtler & Kilger 2002). For each level, it entails defining practical ways to set objectives and efficient ways to assess performance and measure progress (Schwarz et al. 2002). Furthermore, The inclusion of a mechanism or system that monitors, manages and controls the system in order to meet the sustainability requirements is expected (Herrmann et al. 2014). To this end, the main capabilities required for this mechanism are:

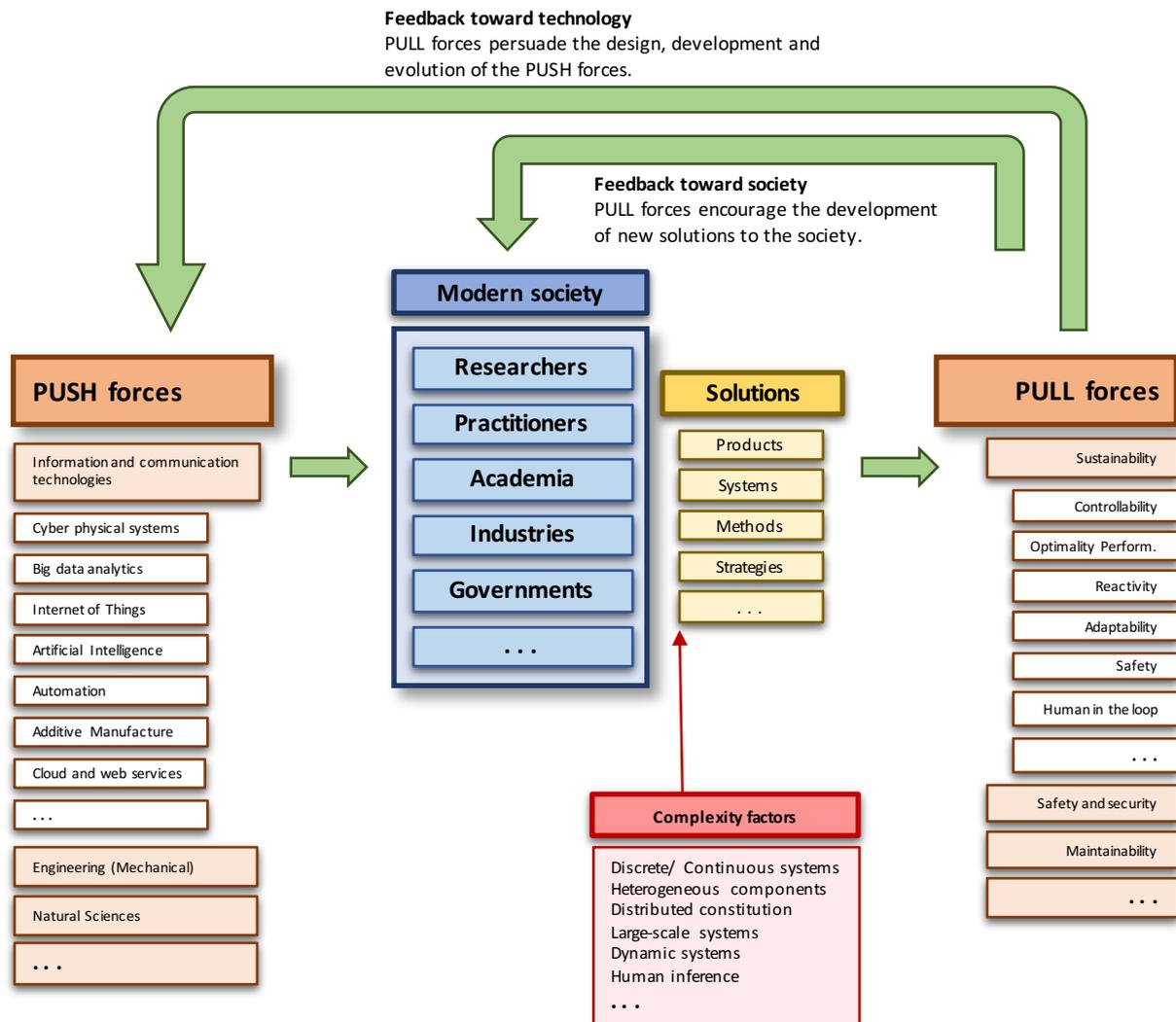
- **Controllability**, as the system must be able to steer a system into an arbitrary final state corresponding to the desired behaviour or functioning (Liu and Barabási 2016)
- **Optimality performance**, as the system must be able to achieve optimal functioning measured by pre-defined efficiency and effectiveness metrics (Kasinger 2010)
- **Reactivity**, as the system must have the ability to act in response to changes within its environment (Kaelbling 1986)
- **Adaptability**, as the system must be able to change behaviour and functioning in response to changes in the environment (Oreizy et al. 1999)
- **Safety**, as it must be able to prevent any danger to human life, equipment or the environment (Zhang & Jiang 2008)
- **Human in the loop**, as the system considers the potential and as-needed interaction of human in any phase of the system (Trentesaux & Millot, 2016)

Modern society imposes its objectives onto the system efficiency and effectiveness goals. The purpose of this imposition is to design the system towards a society pull strategy. Still, this entails addressing many complex challenges, specifically in the design and implementation phase. The complexity factors, which relate mainly to discrete/continuous system interactions, heterogeneous components, distributed control and composition, large-scale systems, dynamic systems and human inference (Seshia, Hu, Li & Zhu, 2016), harden the functioning of the system. However, while the societal pull forces certainly set the direction for the resolution of real world challenges, these complexity factors are key guidelines to consider when constructing a solution.

### 1.2.3 ICT: a contribution to solve the societal needs

Advances from the ICT revolution have the potential to address some of the biggest challenges of society (NIST – U.S. Department of Commerce, 2012). Recent developments have had a profound impact on the development of the evolution of global society (Gunasekaran & Ngai 2012). However, technology must be adjusted to human needs, rather than adapting human action to technological constraints (Herrmann et al. 2014). Technology must meet the challenges of societal needs to fulfil the expectations of humanity. On the one hand, technology has the potential to bring to market cutting-

edge advances for a number of problems (Liu & Zhang 2015). On the other hand, society and new global expectations have a complexity that pulls in the direction of new and flexible solutions to develop a better future (Ueda et al. 2009). Modern society must synchronize its efforts to connect and orchestrate steps for a better future. In this sense, closing the gap between the ICT push forces and the societal pull forces is perhaps the main goal of modern society. Consequently, systems must set their own objectives in this direction. Figure 1 illustrates the fulfilment of societal needs through technological advances.



**Figure 1** Fulfilment of societal needs through technological advances

As illustrated in figure 1, the relation between the pull and push forces is a closed-loop cycle. On the one hand, modern society is encouraged to fill the gap between the technological advances and the needs of the public. Then, overcoming the complexity factors (e.g. discrete/continuous systems, heterogeneous components, etc.), it uses products, systems and other approaches to analyse and create solutions that resolve or contribute to closing this gap. On the other hand, the adoption of solutions and the meeting of global challenges induces feedback from actors in both the field of technology and the society. It then informs the design, development and evolution of new technological advancements,

and it encourages new solutions for modern society. Certainly, the closed-loop cycle is self-powered by the constant feedback received for the enhancement of the push forces and the improvement in the creation of adequate solutions to societal-needs. The following section focusses on the solution of the system modelling and control for discrete-event systems. This is recognized to be a solution that represents the real system, provides a better methodological understanding of the system, promotes identification of the key aspect of the modelling, and employs a systematic approach to develop a solution for the corresponding problem or opportunity.

### 1.3 Theory and control for discrete-event systems

#### 1.3.1 System theory

Sokolowski and Banks (2012) define a system as a “collection of different elements that together produce results not obtainable by the elements alone. These elements may be people, hardware, facilities, political structures, documents or any of the things required as qualities, properties, characteristics, functions, behaviour, and performance”. Jennings (2000) presents a shorter definition emphasizing the purpose of the system, defining “network of components created to solve problems and these are beyond their individual capabilities”. Sayama (2015) includes in the definition the expected results of systems: “a system is a network made of a number of components that interact with each other, may evolve through self-organization and permit the development of an emergent behaviour at macroscopic scales”. Although these definitions define the system concept, the definition by (Whitehead, Boschee & Decker 2012) is used in this dissertation, as it includes the most important parts of the previous definitions: “a system is an organized, problem solving and purposeful structure regarded as a whole that consists of interrelated and interdependent elements (components, entities, factors, agents, members, parts, etc.), which are continually influenced by each other (directly or indirectly) to maintain their activity and the existence of the system to achieve the common purpose or a goal”. In this dissertation, the choice of this definition is predicated on the need to use the system and the system modelling as a problem-solving method that employs different elements to achieve the assigned purpose.

Systems can be divided into two types: natural systems and human-made systems (Wallner 1999). The respiratory system of the human body or an environmental ecosystem are natural systems, for example. On the contrary, the functioning of an emergency room in a hospital or the operations of the New York airport are human-made systems. Both are composed with a set of elements that interact with each other for a specific objective. However, some characteristics, such as the quantity of elements, interaction between elements, or possibility of unpredictable events, imply a certain level complexity that requires different types of analysis to understand the system’s behaviour (Åström et al. 2001). This dissertation focusses on human-made systems, as these are built on computer devices to collaborate or manage certain physical systems. Human-made systems, as seen in the term itself, are systems that are created by humans to plan and/or carry out a desire purpose.

A system is composed of elements, interactions, actions, diversity/variability attributes, environmental characteristics and collective activities (Bar-Yam 1997). These characterize the attributes of the system, its internal functioning and its functioning with its surroundings. The elements are the heterogeneous participants in the system in which a defined internal behaviour may influence the system in a direct or indirect manner. The interactions are the form of relationship between these elements and these determine the form of association featured within them. The formation/operation

actions describe the type of arrangement or structure present between the elements and the functioning which describes each element's behaviour. The diversity/variability attributes are the irregularity of elements and interactions that make the system more erratic and unpredictable. The environmental characteristics, as an element of the system, include the atmosphere where the system is contained, which supports its operation has its own demands and requirements. Finally, the collective activities are the potential actions that can be performed cooperatively and whose parts are generally guided by a specific objective or objectives. In general, the interaction of these parts describes the behaviour of the complex system and the dynamic progression over time.

Two general behaviours can be identified as a result of the system's operation (Kasinger 2010; Serugendo et al. 2006). The first is identified as *emergence behaviour* from the interaction and dynamism of its elements. This emergence is the appearance of a non-trivial relationship and behaviour between elements as a result of several interactions within the system (Sayama 2015). Certainly, in human-made systems emergence behaviour is the result which must be aligned to purpose of the system. On the other hand, it may also be identified as *self-organization* of the structure and behaviour of the whole system. Self-organization is a dynamic and adaptive process where the system acquires and maintains an organized structure on its own without external input (Dias-Ferreira et al. 2014; Picard et al. 2009). These two results maintain and sustain its functioning in an uninterrupted process through time and involve the achievement of the system objectives. For this reason, The analysis of the static and dynamics characteristics of these system is interesting (Åström et al. 2001).

Certainly, systems are challenged by the associated complexity. Complexity is defined as intricately coordinated events or parts that are difficult to analyse, understand or explain (Merriam-Webster Inc. 2004). However, in systems theory, complexity can be defined and measured in different ways depending on the domain and discipline (Elmaraghy, Elmaraghy, Tomiyama & Monostori 2012). For instance, in manufacturing, complexity can be associated with the design of the products or machines, the operations performed in the factory and/or the combinatorial decision-making during production execution (Chryssolouris et al. 2013). Nevertheless, one can identify certain particularities related to the complexity of any type of system. Complexity can be classified according to its static and dynamic characteristics, as both types of complexity are challenges related to the understanding of systems. Static complexity refers to the structure and arrangements defined in the system, the number of elements, the scale of the elements, the interconnections between the elements, the interdependencies and the diversity between the elements. Dynamic complexity refers to the behaviour of the system, the uncertainty and the unpredictability (Chryssolouris et al. 2013).

### 1.3.2 Control systems

A main concern of any system is the need for a control system, i.e. a separate system that monitors and guides the functioning of the overall system towards the achievement of its own objectives. A **control system** is generally known as an arrangement of components and methods that manages the behaviour of another system. The concept of control refers to the definition and organization of a feedback loop interaction between two systems: the controlled (often physical) system and the control system (Trentesaux 2002). The *controlled system*, is a first subsystem with a defined purpose that is managed and guided to achieve its own objective. Some examples of controlled systems are physical, biological, engineering, computational and software systems. The *control system*, as a second subsystem, is often an interconnection of components (i.e. sensor, process/plant, actuator and controller) with the general

objective of influencing the behaviour of the controlled system in a desired way (Jelali 2013). A classic example of a control system is found in industrial automation. Traditionally, control systems are used to regulate or improve the effectiveness of a controlled system (“doing the right thing”). However, a modern view of this concept leads to consideration of effectiveness and efficiency indicators as well (“doing the thing right”) (Baños et al. 2011).

To address certain key concepts in this definition, more can be said about this concept. A control system is a system, thus it can be characterized from an external or internal point of view (Trentesaux 2002). Externally, a control system is characterized by the objective or purpose of the system and its interaction with the environment. Internally, the control system is described through its structure (or arrangement), behaviour (or functionalities) and the dynamics during execution (or progress). This characterization approach corresponds to the understanding of the system according to its composition (what is it?) and its actions (what do we do with it?) (Lemoigne, 1994). This method for describing a control system serves as a breakdown of the concept and provides a general picture of the scope of this term. Figure 2 illustrates the nature of a control system and its characteristics.

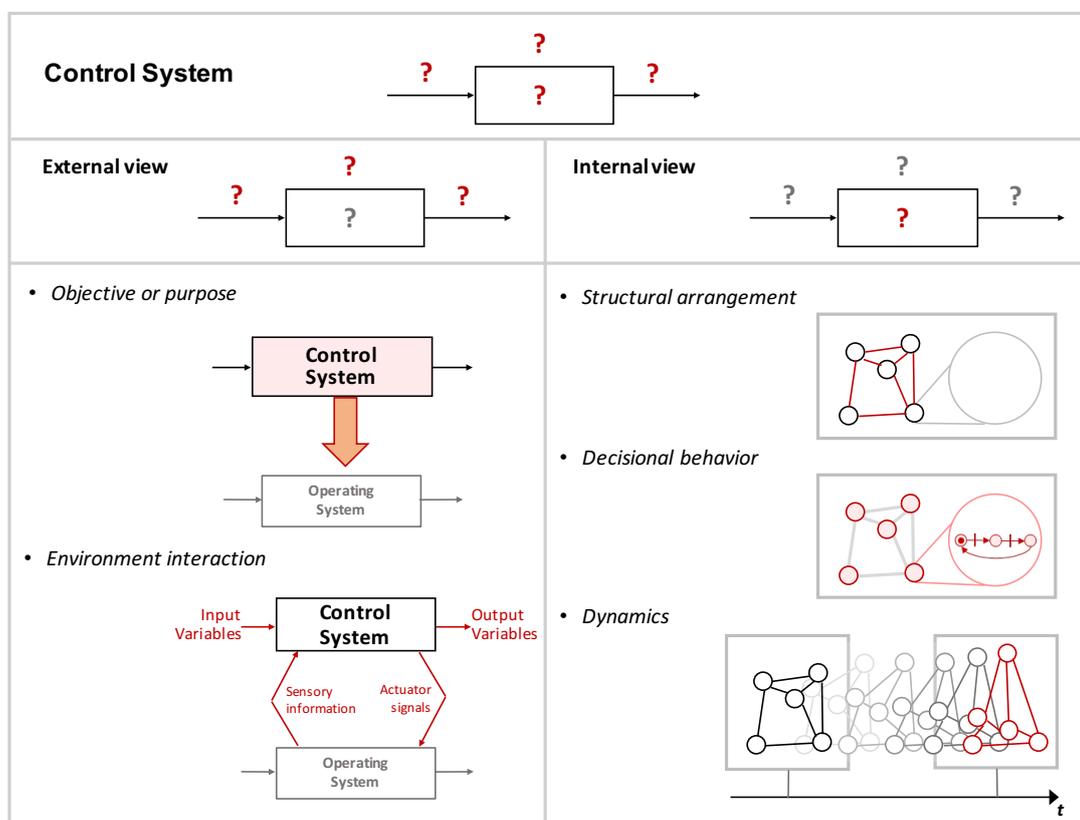


Figure 2 Nature of a control system and its characteristics (adapted from Trentesaux (2002)).

### 1.3.2.1 External view of a control system

An analysis from an external point of view considers the system as a black box. The system is thus analysed in terms of the *objective or purpose*, and the *environment*.

- **Objective or purpose:** the purpose of the control system consists in guiding the activities of a controlled system in order to accomplish pre-established objectives (Trentesaux 2002). One of the most important driving forces or purposes of any system is related to the efficiency and effectiveness indicators (Gunasekaran & Ngai 2012). These two concepts are commonly referred

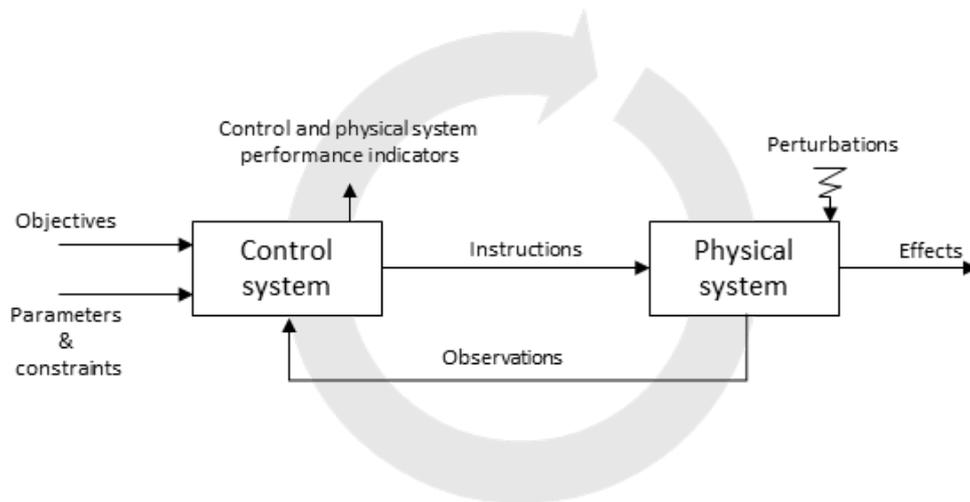


Figure 3 Model of the environment in a control system (Trentesaux 2002).

to as the key performance indicators used in different domains and set the course of action to properly control the physical system (Giret et al. 2015).

- **Environment:** The environment of the control system relates to the interaction between the system and the external world. The control system continually monitors the situation of the environment and refines a response based on changes (Sokolowski & Banks 2012). This is because the environmental context can affect the reactivity and behaviour of the control system (Nof 2009). Figure 3 presents a model of the interaction between these two systems: control system and physical system, based on the fundamental models of control theory (Trentesaux 2002).

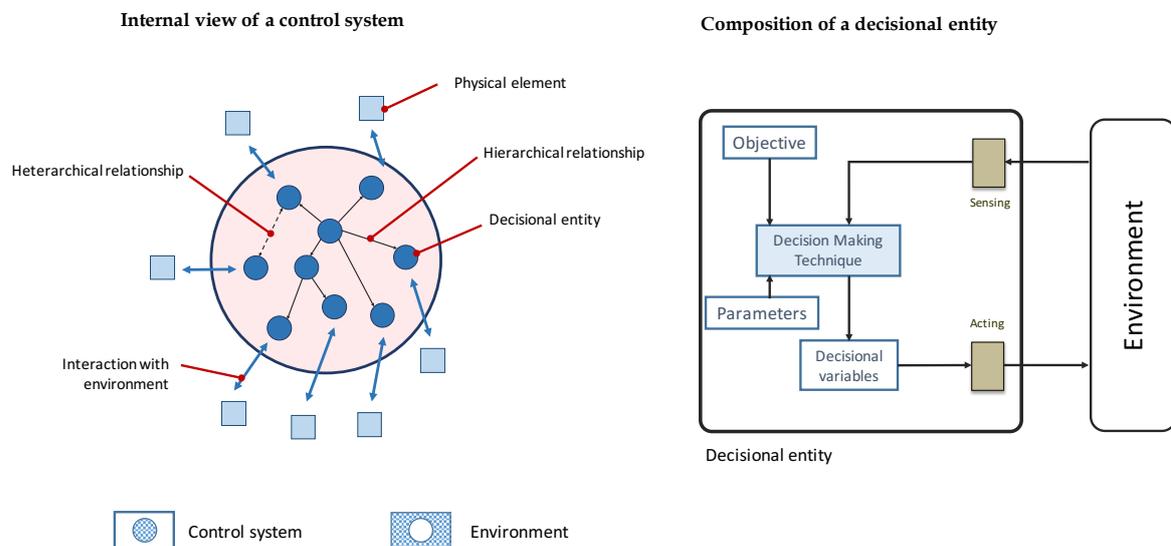
The model is from the classic control perspective (Baker 1998), as it represents a closed-loop system where observations of a controlled system (i.e. physical system) provide input to the feedback process to activate the steering of the control system through instructions or commands. The inbound information is the objectives (intended results for both systems), parameters and constraints (boundaries and degree of movement), and the observations from the monitored behaviour. The outbound information is the instructions for influencing the controlled system and the performance indicators regarding both systems.

### 1.3.2.2 Internal view of a control system

The internal view of the control system is defined by the architecture of the control system, referred to as the control architecture. In the literature, the term control architecture tends to be used to describe the composition and functionalities of the control system (Senehi & Kramer 1998). It encompasses the specifications of the structure, behaviour and dynamics that define the functioning, describe the progress, and set the rules for the emerging actions in the execution of the control system. The control architecture is a critical issue in the design phase as, from a systematic point of view, it is the input of

the control system whose output is the performance with regard to the regulating activity (ElMaraghy 2008). The control architecture consists of the components, the structure, the behaviour, and the dynamism of the control system.

- Components:** Control systems consist of a set of hardware devices and/or computer software that interact to solve the system's problems or meet its objectives. One of the major difficulties of control systems is to simultaneously handle the complexity and real-time requirements associated with the decision-making of a controlled system. For this purpose, control systems divide the entire control problem into several components to distribute the decisional capabilities (Trentesaux 2009). The process of a control system is definitively decisional (Dilts, Boyd & Whorms, 1991). Therefore, these components are decisional entities with goal(s), parameters and restrictions that can sense some or all of the physical system, execute the decision-making process and trigger certain instructions to the controlled system. In this dissertation, a **decisional entity** is an autonomous unit able to communicate, make decisions and act within a particular scenario (Duffie 1990). The decisional entity is a subsystem that is able to support a decisional process and is composed of a monitoring activity, problem formulation, problem solving and action executed through the corresponding actuators (Trentesaux 2009). In detail, it contains a decision-making technique that helps select a determinate solution from among several possible alternatives (Debenham & Prodan 2013). To this end, each possible alternative is evaluated according to the specific objective or objectives and, in a decisional problem, a course of action is chosen. Thus, the decision-making technique drives the behaviour of each component. Figure 4 illustrates internal view of a control system and the composition of a decisional entity.



**Figure 4** Internal view of the control system and composition of a decisional entity

- Structural arrangement:** The structure refers to the organization or arrangement of the component in terms of its architecture, which is generally defined through the types of communication or relations between them. These relationships can be classified into two types (Trentesaux, 2009): hierarchical and heterarchical relationships. A hierarchical relationship occurs when two components have a master-slave relationship where one fully dominates the actions of the other. Conversely, a heterarchical relationship is present when two components have a

cooperative relationship in which they collaborate and share responsibility for executing their actions and the achieving their own objectives together. The aggregation of these pairwise relations between components constitutes the structure of a control system. Considering the possibilities of this aggregation, four elementary types of control system structures are generally used in the literature: centralized, fully hierarchical, fully heterarchical and hybrid/semi-heterarchical structures (Trentesaux, 2009). Each structure features different characteristics in terms of information availability, global (e.g. whole system) and local (e.g. specific components or subsystems) decisions, responsibilities/objectives, metrics/performance, allocation of decision-making, autonomy of components, impact on the network, fault tolerance and dynamic adaptability control. According to the control objective, these arrangements may be used to distribute and allocate the global control problem in a network of decisional entities. This disaggregation serves to reduce complexity in the overall problem, diminish the uncertainty and hierarchize in order to prioritize certain objectives (Stadtler & Kilger 2002). Certainly, the choice of control system structure to manage a controlled system depends on the implementation requirements. Figure 5 illustrates and explains the characteristics of each approach.

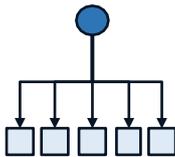
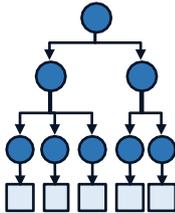
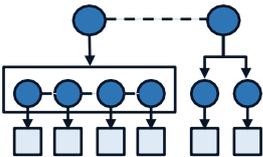
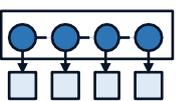
	Centralized	Fully Hierarchical	Semi Heterarchical	Fully Heterarchical
Structure diagram				
Description	The control is allocated to a single component in a coercive and hierarchical relationship.	The control is divided between a set of components that are connected in a hierarchical relationship.	The control is divided between a set of components that are connected either in a hierarchical or heterarchical relationship.	The control is divided between a set of components that are connected in a heterarchical relationship.
Purpose	A single component has acknowledge and autonomy over the whole system	Autonomy and decisions are distributed in several layers to diminish the complexity	The control is divided in different types or relationships in order to tailor the needs of the system.	Components share the autonomy to impulse system reactivity and collaboration

Figure 5 Types of control system structures (Adapted from Dilts, Boyd, and Whorms (1991))

- Behavioural functioning:** the behavioural functioning of the components is guided by the decision-making technique of the decisional entity. There are several types of decision-making techniques in the literature, including protocols (Borangiu et al. 2013), priority rules (Szelke & Kerr 1994), heuristics (Tan & Aufenanger 2011) and metaheuristics (Voß 2001), to name a few. These techniques can be classified according to the associated complexity of the decision process. Each component performs a specific algorithm that evaluates the alternatives and selects a solution according to a predefined objective. The associated complexity depends on the amount of information that it is required, the scope of the objective and the computational effort needed to reach a solution (Chryssolouris et al. 2013). In addition, these also determine the time of execution needed. In control systems, the decisional entity is classified based on the degree of optimization and the execution time needed during the decision process. On the one hand, the degree of optimality in the approach can be *optimal* (e.g. mathematical programming, combinatory optimization), *near-optimal* (e.g. approximation algorithms, probabilistic algorithms), *towards-optimal* (e.g. artificial intelligence, neural networks, evolutionary algorithms, swarm algorithms)

or *heuristic* (e.g. artificial intelligence, priority rules, iterative simulation) (Baker 1998). On the other hand, the execution time is the time required to return a solution, which usually depends on the number of feasible alternatives, the number of evaluations performed and the solution search process (Rardin 1998).

Several studies have classified the decision-making techniques on the basis of various characteristics, such as the optimality achieved and the use of deterministic/stochastic parameters. This study uses the classification proposed by Chaari, Chaabane, Aissani & Trentesaux (2014). The decision-making techniques are divided into predictive techniques, reactive techniques and predictive-reactive techniques.

***Predictive techniques.*** These techniques are generally based on operational research techniques and exact methods to address an assigned problem or objective, and generally seek global optimality of the system. The main characteristic of these approaches is that they provide an optimal or near-optimal solution of a control problem. Still, these approaches assume extensive availability of processing time, full availability of information, reactivity components and optimal system behaviour. However, the lack of reactivity is one of the main disadvantages of these approaches. For this reason, the execution of predictive approaches is mainly, but not exclusively, used for the activities from offline processes (i.e. those carried out before execution). Some examples of these approaches are mixed-integer linear programming, non-linear programming and genetic algorithms (Neungmatcha, Sethanan, Gen & Theerakulpisut, 2013; Pach, Berger, Bonte & Trentesaux, 2014; Zambrano Rey, Bekrar, Trentesaux & Zhou, 2015).

***Reactive techniques.*** These techniques are generally based on artificial intelligence or heuristics methods. The main characteristic of these approaches is that they provide a very fast responses when required. However, this short execution time in the decision-making process prioritizes the speed rather than the quality of the response. In this sense, it does not feature optimal characteristics. Thus reactive approaches are mainly used for online/real-time processes (i.e. during execution). Examples of these approaches are priority rules, neural networks and greedy algorithms (Bekkar et al., 2016; Hegde & Gray, 2017; Zahmani, Atmani, Bekrar & Aissani, 2015).

***Predictive-reactive techniques.*** These are generally hybrids that use both predictive and reactive techniques simultaneously. These approaches are designed to benefit from the advantages of predictive and reactive techniques and limiting the associated drawbacks (Thomas, El Haouzi, Klein, Belmokhtar & Herrera, 2009). In general terms, these techniques have two phases during system execution: offline, or the system setup time, and online, or the time after execution has started. While the predictive techniques are generally used in the offline phase to set certain global instructions towards global optimality, the reactive techniques are used during the online phase to support the system operation in response to perturbations or uncertainties. An example of this approach is proposed by Zambrano Rey et al. (2014b), in which a predictive genetic algorithm is coupled with a reactive distributed arrival-time control-priority for a flexible manufacturing system. Other examples can be found in Baños et al. (2011) and Chu, You & Wassick (2014).

- **Dynamic progression:** Dynamic progression, which refers to the motion of the control system, is the actions and progress that take place during system execution. Throughout this dissertation, this term is used to refer to the interactions between the components and the variation regarding the structure and behaviour of the control architecture. Configuration of the control system must then

involve establishment of the strategy and policies that the control architecture follows in order to encourage a specific dynamic development. Certainly, the strategy and policies must be aligned with the structural arrangement and the behavioural functioning as defined previously. However, these also encompass the aggregated interactions of the components and the functionalities of all components operating in the controlled system and the control system (Frei, Barata & Onori, 2007). The dynamic progression can be classified in three levels: self-organization, adaptation and evolution. Self-organization is the ability of a system to change its structure or behaviour dynamically and spontaneously in response to changes in conditions without external intervention (Barbosa, Leitao & Pereira, 2011). Adaptability is the ability of a control system to change its architecture in order to improve performance over a period of time (Bordoloi, Cooper & Matsuo, 2009). Evolution is a long-term property where the system overcomes disturbances through changes (Neves et al. 2011) and optimizes its attributes over time for its own survival (Sayama 2015). Each of these levels has the potential to achieve the expected efficiency, effectiveness and reactivity of control systems (Belisario & Pierreval 2013). In general, these is created to respond to any perturbation that occurs during execution.

### 1.4 Configuration of control architectures

#### 1.4.1. From static configuration of control architectures...

The control system configuration refers to the manner in which the components of the system (or sub-system) are chosen and put together for control system execution (Pierreval & Paris 2003). ElMaraghy (2008) builds on this definition by implying that this configuration refers to the description of the control structure and various functional units, where the outcome of this process is needed in a specific application. In general, these definitions specify the concept of configuration as a selection of the parameters or components of control architecture for the achievement of a specific purpose.

In this dissertation the concept of ‘configuration of the control architecture’ is used in its broadest sense to refer to the architectural composition of the control system, including the internal composition of the decisional entities (**components**), the selection between fully hierarchical, fully heterarchical or semi-heterarchical arrangement (**structural arrangement**), the parameterization of the decision-making techniques to be executed by the decisional entities (**behavioural functioning**), and the definition of the interactions between the decisional entities (**dynamic progression**) that aim to achieve one or more objectives (**Purpose**) during execution within a particular atmosphere (**environment**).

The process for the configuration is a necessary but challenging task for the objectives of the control system. Generally, this process is an iterative evaluation of several combinations that aim creating an architecture that achieves a specific objective (Alencar & Lucena 1996). Indeed, this process is critical, as the chosen configuration impacts the performance of the system and represent the expected goals to meet these requirements. Depending on the chosen configuration, the control system may then reach the desired efficiency, effectiveness and reactiveness levels differently. For this reason, the selection of the configuration is seen as an optimization problem based on different characteristics, e.g. neural network training (Mouelhi-Chibani & Pierreval, 2010) or quality attribute criteria (Sanchez, Diaz-Pace, Zunino, Moisan & Rigault, 2011).

An optimal configuration of the control architecture is the key driver for good system performance (Koren, Hu & Weber, 1998). In addition to considering the specific tasks (i.e. purpose) required to steer the control system, this optimal configuration depends significantly on the structure (i.e. fully hierarchical, fully heterarchical or semi-heterarchical), the behaviour of the component (i.e. predictive or reactive) and the dynamism progression (i.e. emergence) that are defined. In operational research, the wide range of possible combinations for the configuration of control system concerns a set of optimization problems called combinatorial optimization (Blum & Roli 2003). This problem refers to the choice of a “best” configuration from a set of alternatives to maximize or minimize under certain constraints. Considering this challenge, the combinatorial techniques for configuring control systems represent a good approach to improve system efficiency and effectiveness. Examples of these techniques include network configuration, expert systems and multi-objective optimization algorithms (Badran, Mekhilef, Mokhlis & Dahalan, 2017; Filip, 2008; Xiang, Fan, Liu & Wang, 2010).

The configuration of a control architecture requires analysing the arrangement of multiple interrelated components. The configuration theory argues that the selection of the configuration depends on the best-suited contextual conditions (Bedford & Malmi 2015). Nevertheless, the combinatorial possibilities of the possible arrangement and contextual conditions suggest addressing these problems to achieve the “best” configuration through an optimization approach. In the literature, there are two approaches to configuring the control system (Senehi & Kramer 1998): the *evaluation/assessment approach* and the *simulation-optimization approach*. The *evaluation or assessment approach* uses the characteristics of the configuration to assess the performance of the control system. Some examples of these approaches can be seen in energy control configuration (Moutassem & Anders 2010) and supply chain configuration (Graves & Willems 2005). The *simulation-optimization approach* uses an offline simulation of different alternatives, enabling an evaluation of the behaviour during execution. Some examples of this approach include simulation-optimization for manufacturing control (Zambrano, et al. 2014a), machine learning for the gas industry (Hegde & Gray 2017), and Tuning the multi-agents behaviour (Bernon et al. 2007). Certainly, both approaches obtain a solution striving for an optimal configuration and represent a structured strategy to set the control architecture. However, the most important aspect is that both can be framed throughout optimization-based principles that attain to best configuration. These are: representation of alternatives, evaluation of alternatives, iterative research process, stopping criteria, and criteria for solution creation. A detailed description of the optimization-based principles is found in Appendix A

The main challenge in the configuration is that, during execution, the defined configuration does not provide the same performance that was initially intended. Although the control system can be configured to respond to a wide range of changes and perturbations, it may be limited when acting in unforeseen situations (Jimenez, Bekrar, Trentesaux, Montoya-Torres & Leitao, 2013). This problem arises from the changes and evolution of the highly volatile environment in the controlled system (Neves et al. 2011). In this regard, researchers and industrialists have worked on addressing this problem in various ways, including predictive-reactive approaches (Vlk & Barták 2015), re-\* techniques (rescheduling, re-planning, re-routing) (Katragjini, Vallada & Ruiz 2015), dynamic scheduling (Madureira 2001), and many others. These largely intend to recover the initial execution in response to the specific changes that occurred in the system. For instance, rescheduling approaches aim to recover the initial decision-making either to sustain the same global performance or minimize the performance degradation (Vieira, Herrmann and Lin 2003). Nevertheless, although these approaches promise advances in the reactivity to environmental uncertainties, further research must be done to work with the dynamism associated with these systems (Brennan, 2016).

The approaches with this limitation are known in the literature as static control architectures (S-CA). In S-CAs the control system starts with the initial configuration of the control architecture and maintains the same structure and behaviour throughout the execution process (e.g. online phase). S-CAs have certain advantages and disadvantages regarding the performance of control architectures (Cardin et al. 2015; Zambrano Rey et al. 2013). The advantages of the S-CA are that the control system can achieve optimal performance and can be built on a simple and dedicated software, and the control system is usually able to achieve the expected objectives. However, the drawbacks of the S-CA are that the control system cannot respond optimally to any change during execution, the configuration assumes full information on the environment (before and after execution), and a potential perturbation during execution will alter and degrade the expected result. These disadvantages of S-CAs suggest that there may be a need for control architectures to be configured both at the beginning of execution and during the execution process. Perhaps, a change of configuration will maintain the expected performance during execution.

### 1.4.2. ...to dynamic configuration of control architectures

Recent years have seen a growing trend towards configuration of the control architecture during execution (Manceaux, 2015). This new approach, here referred to as dynamic control architecture (D-CA), contributes to handling and managing the events taking place during the execution of the control system. In broad terms, D-CA changes the structure and behaviour of the control architecture in response to the system requirements. The control system starts with an initial configuration which then changes over time in order to provide the functionality and capacity needed at the time when is needed (Youssef & ElMaraghy 2007). Some examples of this new approach can be found in (Cardin & Castagna 2009; Leitão & Restivo 2006; Van Brussel et al. 1998). Then, when a perturbation occurs, the control system aims to maintain its performance during execution and recover to the initial expected performance by executing adequate changes to the control architecture (Terkaj, Tolio & Valente, 2009). In brief, D-CA approaches are likely to be able to tailor the control solution to tackle the complexity and uncertainty of any system. A reasonable approach to improving the efficiency, effectiveness and reactivity of control systems could be the inclusion of a mechanism that steers their changes and dynamic development. However, this statement must be interpreted with caution, as further research in this area is needed, and certain requirements and challenges must first be resolved in order to maintain the required performance throughout the change of the control configuration.

As previously stated, the reconfigurability characteristic of D-CA approaches attempt to maintain the efficiency, effectiveness and reactivity of control systems. The concept of reconfigurability is defined as the ability to repeatedly change and rearrange the components of a system in a cost-effective way (Setchi & Lagos 2004). The main benefits of implementing reconfigurability features in control systems are the ability to customize the control architecture in order to respond to different system requirements, improved reactivity of the system, and increased robustness and resilience of the system in the case of uncertain events (Lyke, Christodoulou, Vera & Edwards, 2015). However, these approaches must first resolve the following issues in order to return to the expected control system: how flexible should a control system be? When should a configuration change be triggered during execution? Which characteristics, parameters or protocols should be altered to provoke a configuration change? How can it be assured that the new configuration is suitable (or optimal) for the control system? And how should the new configuration be synchronized with the

ongoing execution? While the reconfiguration of control architecture still has a considerable way to go to ensure maintained efficiency, effectiveness and reactivity performance, the resolution of these questions will definitively contribute in this direction. In fact, in addition to this contribution, the inherent characteristics of D-CAs suggest a potential solution to the gap between the technological push forces and the societal pull forces. On the one hand, it offers the flexibility needed to adopt different technological advances from the ICT revolution. On the other hand, it can tailor the configuration to fulfil certain conditions and objectives. Certainly, D-CA has the potential to close the gap between these pull and push forces. For this reason, the remainder of this dissertation focuses on the D-CA approach. In this dissertation, the terms “D-CA” and “reconfigurable control system” are used interchangeably to mean any control systems that change their control architecture configuration during execution.

### 1.5 Summary

This chapter presents the topic of control systems as a key topic for addressing many global challenges. To this end, it presents the context of the functioning of control systems and highlights the need for further research on approaches that dynamically change the control architecture to improve the efficiency, effectiveness and reactivity of control systems. This contextualization first addressed the progress of the technological revolution with the current needs. This chapter also presents the opportunities for using these informational and computational technologies, e.g. cyber-physical systems, automation, and the internet of things, to address pressing global challenges. Certainly, there is an interrelation between the push forces of technological progress and the pull forces of societal needs. Then, as a key domain that articulates the use of ICT technology, a descriptive characterization of control systems was presented from an external/internal view, and, simultaneously, various alternatives to instantiate such systems were presented. Certain key concepts related to the configuration of control architectures and the static/dynamic control architectures were presented. Clearly, the solution of the control system depends directly from the control architecture. Still, the static control architectures are partially covered from possible perturbation because they addresses with reactivity only for low levels. However, dynamic control architectures perform better as it provides both high and low-level reactivity. The following chapter focusses on reviewing the most relevant literature related to reconfigurable control systems, gathering insight regarding system architectures, and examining in detail the functioning of the reconfigurable process.

# Chapter 2

## Reconfigurable control systems: literature review

### 2.1 Introduction

The previous chapter presented a contextualization of discrete-event control systems. It described the composition of control architectures, highlighted the main characteristics of static and dynamic control architectures (S-CA and D-CA), emphasized the need for further research in D-CA (reconfigurable control systems), and suggested a means to improve the efficiency, effectiveness and reactivity of control systems. The objective of Chapter 2 is to review the literature on reconfigurable control systems and characterize the most relevant contributions of this approach. This chapter aims to study these contributions, examine in detail the functioning of these architectures, determine how they are used to improve the efficiency, effectiveness and reactivity of control systems, and point out some limitations in this approach. This characterization leads into the contribution presented in Chapter 3.

In order to present an exhaustive characterization of the existing approaches to reconfigurable control systems, Section 2 presents a theoretical framework through the exploration of important concepts in reconfigurable control systems (RCS) as concerns the purpose and the means for its achievement. Section 3 undertakes the literature review and provides the insights of existing RCS that feature an optimality initiative. This section characterizes the existing approaches and analyses the contributions with regard to classifying the functioning of the reconfigurable process. Finally, a summary of the chapter is presented in Section 4.

### 2.2 Theoretical framework of reconfigurable control systems

This section presents a theoretical framework that serves as a baseline to define the key concepts for positioning and evaluating the RCS literature. Generally speaking, RCS are control

systems whose architecture can change according to the control system needs (Martin & Barber 2006). Figure 6 illustrates the relationship between the RCS and the societal needs.

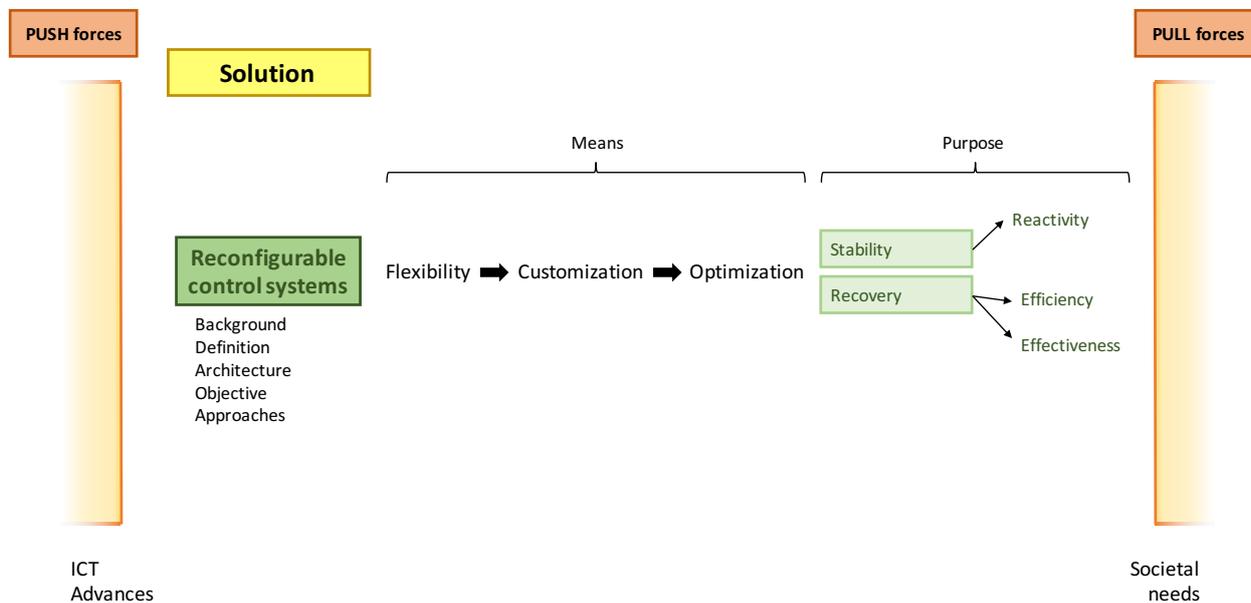


Figure 6 Reconfigurable control systems within the context of ICT as a solution to meet societal needs.

### 2.2.1 Reconfigurable control systems: technological solution

This subsection presents certain concepts related to reconfigurable control systems, including background, definitions, architecture composition, objective/purpose and modelling/classification approaches.

**Background:** Generally speaking, reconfigurability is not a new concept in the ICT domain. As explained in the previous chapter, reconfigurability is a concept that can be applied to the control system itself. The exact origin of the RCS is not clear, but it is believed that it mainly lies in the reconfigurability of hardware, more specifically in reconfigurable computing (Setchi & Lagos 2004). In 1960s, Estrin proposed a standard main processor that, when augmented with an array of reconfigurable hardware, controlled the execution a different behaviour according to the task for which the hardware was set up, such as image processing or pattern matching (Estrin 1960). Therefore, RCSs were based on the changeability of interconnection and routing between components. Later on, the concept of RCS was enhanced in 1980's by considering several interfaces for the component in Reconfigurable robotics. RCS have not only reconfigurability in the interconnections but also modularity characteristics from the system components. Beginning in the 1990s, the development of field-programmable gate arrays (FPGA) required reconfigurability features in order to customize different routings and components (e.g. logic blocks) of the circuit for customer requirements. The functionality of the connections and the logic blocks of the network can be modified by downloading bits of configuration data, known as a bit-stream, onto the hardware (Bondalapati & Prasanna 2002). These systems, called reconfigurable computing systems, required a combination of hardware and software flexibility to control the functioning of the FPGA's components. In addition to reconfigurable computing, RCS and this research area are enriched by developments in control theory, adaptive control, software architecture, among many other disciplines.

**Definition:** The definition of RCS has evolved over time. Definitions change according to the functioning and the field of application (Jiang and Yu 2012). Several definitions are currently proposed. For Medvidovic (1996), RCSs are control systems able to change the configuration of the control architecture during runtime. Batchkova et al. (2013) define RCS as a “control system able to rearrange dynamically the elements of the control architecture to accommodate to failure events and new requirements”. Konstantopoulos and Antsaklis (1999) proposed the following definition: “control systems that are characterized to perform under the presence of drastic changes in the system dynamics – such as changes in the operating conditions – and are responsible for guaranteeing a stability in the functioning and recovering maximum control performance under impairments”. Clearly, these definitions state that RCSs are based on modification of the elements or components within the system, specifically elements of the control architecture. However, there are other approaches that address the definition of RCS as changes in the control system’s purpose and interaction with the environment (e.g. control objective, policy, strategy, etc.) rather than changes in the architecture. For example, Jiang (1994) states that RCSs are control systems capable of dealing with variations in the environment, but considering the dynamic rearrangement based on the adjustment of the control strategy. Simon, Kováčsházy and Péceli (2002) add that RCSs are control systems designed to react to failures by accommodating changes in the control objectives. However, these latter approaches fail to fully define the concept of RCS, as the reconfigurability featured is often considered in isolation for a set of control inputs and controlled outputs, without regard to integration into an overall logic-based decision making process (Lincoln et al. 2010).

**Architecture:** The features and benefits of RCSs are based on the changes of the control architecture (D-CA), also referred to as reconfigurable control architectures (RCAs). As explained in the previous chapter, RCAs have two main characteristics: on the one hand, they are control architectures with the capability of changing their structural arrangement and behavioural functioning when needed. On the other hand, they are able to tailor the configuration of the control architecture to handle the complexity and uncertainty requirements. For achieving these two characteristics, the RCS contains a reconfiguration mechanism that changes and tailors the configuration of the RCA. The reconfiguration mechanism executes a reconfiguration process as a method to evaluate the needs of the control system and implement the specific changes to implement a change of configuration. However, in order to execute this reconfiguration, the components of the control architectures have certain characteristics that permit this changeability within the architecture. A precise list of characteristics is elusive, as there are multiple approaches in the literature. For example, Koren et al. (1999) proposed that RCA components must have the following core characteristics: modularity, integrability, customization, convertibility, scalability and diagnosability. On the contrary, Onori, Barata & Frei (2006) proposed that the characteristics should be: modularity, granularity, plugability and evolvability. Whatever the differences among researchers, the RCS components should have enough flexibility in the system in order to be able to change both the structure and the behaviour of the architecture. This flexibility and the ability to set this flexibility during execution constitute the main characteristic of RCS.

**Objective:** The general objective of the RCS is to perform the necessary changes to ensure the efficiency, effectiveness and reactivity of the control system. On the one hand, the search for overall global performance defined by the efficiency and effectiveness implies that these are built to ensure the achievement of the system goals in an efficient and effective manner (ElMaraghy 2006). On the

other hand, the reactivity of RCS implies that it is designed to ensure stability in the functioning of the control system and recover control performance under impairments (Konstantopoulos & Antsaklis 1999).

**Approaches:** There are several modelling approaches used in RCSs, such as multi-agent systems (Khalgui, Mosbahi & Li 2011), holonic systems (Barbosa et al. 2015), service-orientated systems (Da Silva et al. 2016), and many others. This type of modelling uses agents, holons or function blocks as components of the control system (Brennan & Norrie, 2001). This permits to build the control system in a distributed manner and include the reconfigurability of the RCS. The benefit of this modelling is that it responds to optimality and customization as it features sufficient flexibility to meet the demands. The main advantages of using this modelling are that it permits one-to-one modelling of the components of the controlled physical system (Holvoet & Valckenaers 2007), it allows the RCS to be highly resilient to perturbations due to the control decentralization (Botti & Giret 2008), it eases the changeability of the control architecture (ElMaraghy 2006), and it aids in establishment of intelligent interactions to enhance global performance.

There are several approaches that propose a categorization of reconfigurable control systems. As previously stated, while some approaches classify the RCS with a focus on the characteristics that can be changed (Valls, Lopez & Villar, 2013; Dennis et al. 2014; Walsh et al. 2006), others propose a classification according to the type of reconfiguration in the environmental context (Zhang and Jiang, 2008). Table 1 describes these approaches, taking into consideration both the changes and the expectations within the change. Each approach is evaluated according the type of change, i.e. a change in the structure or the behaviour, and the location of the change, i.e. inter-component or intra-component. Appendix B provides a detailed explanation of these classification approaches.

**Table 1** Reconfiguration approaches in reconfigurable control systems

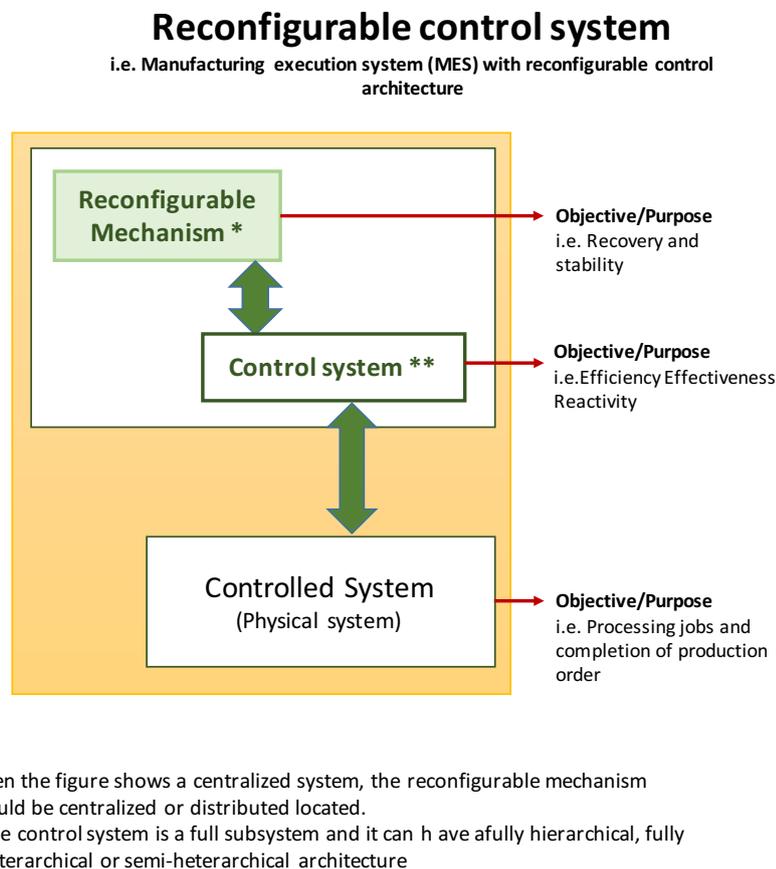
Ref.	Focus of reconfigurability	Categorization	Type (1) and location (2) of reconfiguration
<b>Valls and Val (2014)</b>			
	Reconfiguration based on the combination between different characteristics in both the internal and external view of the control system`	<ul style="list-style-type: none"> <li>a) Software protocols or Performance-based</li> <li>b) Non-real time or Real time</li> <li>c) Centralized or Distributed</li> <li>d) Closed configurations or Open configurations</li> <li>e) Entity execution or System level execution</li> <li>f) Hardware related or Software related</li> </ul>	<ul style="list-style-type: none"> <li>1) Behaviour</li> <li>2) Inter-component, and Intra-component</li> </ul>
<b>Walsh, Bordeleau and Selic (2007)</b>			
	Reconfiguration based on the evolution and adaptation to the new conditions	<ul style="list-style-type: none"> <li>a) Architectural</li> <li>b) Topology</li> <li>c) Substitution</li> <li>d) Protocol</li> <li>e) Interphase</li> <li>f) Internal</li> </ul>	<ul style="list-style-type: none"> <li>1) Structure, and Behaviour</li> <li>2) Inter-component, and Intra-component</li> </ul>
<b>Dennis et al. (2014)</b>			
	Reconfiguration based in the characteristics of the architecture and the applications to the control system	<ul style="list-style-type: none"> <li>a) hardware devices</li> <li>b) low-control functioning</li> <li>c) agent-level</li> <li>d) agent decision-making level</li> <li>e) components interactions.</li> </ul>	<ul style="list-style-type: none"> <li>1) Behaviour</li> <li>2) Inter-component, and Intra-component</li> </ul>
<b>Zhang and Jiang (2008)</b>			
	Reconfiguration based on the strategy and mechanism included in the system	<ul style="list-style-type: none"> <li>a) Optimization</li> <li>b) Switching</li> <li>c) Matching</li> <li>d) Following</li> <li>e) Compensation</li> </ul>	<ul style="list-style-type: none"> <li>1) Structure, and Behaviour</li> <li>2) Inter-component</li> </ul>

Whether describing the RCS through the changeable characteristics or through the expected strategy, both approaches provide a good foundation to categorize the reconfigurable control systems. Such approaches, however, fail to offer a categorization that includes both the changeable characteristics related to the internal view of the system, and the change in purpose and strategy related to the external view. A likely explanation is that these approaches focus on the system composition (internal and external view) rather than the reconfiguration process. Although this is a subject for future research and falls outside the scope of this dissertation, we may assume that a categorization of a dynamic process should be based on the dynamism part of the RCS rather than the static part. Nonetheless, the review of these approaches helps to identify the dynamism characteristics, such as location of changeability, type of changeability and characteristics of the reconfiguration process, which represent the key concepts to evaluate the dynamism of the RCS.

### 2.2.2 Purpose of reconfigurable control systems

As defined above, RCSs are an inter-related composition of systems that act together to resolve a specific global control problem. These systems are (see Figure 7): the reconfiguration mechanism (e.g. reconfiguration sub-system), the control system (e.g. sub-system of the RCS) and the controlled system (e.g. physical sub-system). Each of these systems has its structure and behaviour configured to meet its own objectives. The connection between them is clear because they are related through the reconfiguration control architecture. Nevertheless, it should be noted that the purpose of each is also

interrelated in an escalated manner. For instance, a manufacturing execution system (MES) with reconfigurable control architecture is an example of an RCS. On the one hand, the control system in the MES helps to achieve the purpose of the controlled system (i.e. processing of the jobs) and aims to fulfil the control objectives (i.e. efficiency, effectiveness and reactivity). On the other hand, the RCS helps to achieve the purpose of the control system, also seeking the best configuration.



**Figure 7** General framework of the reconfigurable control system

The purpose of the RCS is to achieve simultaneously the recovery and stability of control systems. Both objectives enhance the performance of the control system and improve the global performance of the controlled system (Rooker et al. 2009). While the recovery is related to improving the efficiency and effectiveness of the system, the stability is achieved is related with the reactivity.

### 2.2.3 Means of reconfigurable control systems

The main advantage of RCSs is that the flexibility provided by the control architecture enables customization and the potential optimization of the control system (Phocas et al. 2015). This theoretical framework highlights the means or characteristics of the RCS that help to carry out the stated purpose (e.g. recovery and stability). This framework focusses on three means: flexibility, customization and optimization.

The first means to achieve the stated purpose is **flexibility**. The most common definitions of flexibility are the ability to take up different positions or, alternatively, the ability to adopt a range of

states in a system (Slack 1983). In the context of this dissertation, the term flexibility is referred to the ability to offer a variety of possible states in the characteristics of the control architecture. Each of the components of the RCA have different states or modules that can be changed during execution. For Walsh, Bordeleau and Selic (2007), the states can be defined according to the structure and behaviour for both the configuration of the component (e.g. intra-component) and the configuration of the architecture (e.g. inter-component). The architecture of the control system can then take a specific configuration depending of the aggregated states of components and, therefore, change the architecture repeatedly as needed throughout distinctive configurations.

Considering the flexibility of RCA, the **customization** of the flexible configuration is a second means that helps achieve the goals of the control architecture. The term “customization” is used here to refer to the capability of the control system to set and adapt a specific configuration of the control architecture in order to meet the specific control requirements (Rivera 2005). Truyen et al. (2001) defined customization as the process of selecting a specific set of characteristics in order to build a more suitable configuration. Customization makes it possible to steer the control architecture as needed and improve the controllability of the system. In the context of an RCS, customization offers the possibility to obtain a proper configuration not only at the beginning, but also during system execution. The RCS can then be tuned, reshaped, or otherwise altered to suit the purposes of users (Lyke et al. 2015). Nonetheless, in the literature, the customization is usually the responsibility of the system designer, and despite some methodological attempts, it remains largely a part of the offline planning phase (Cotta, Sevaux & Sørensen 2008). However, the customization in the RCS offers a wider set of alternatives to respond to the challenges during the online/real-time phase. For this reason, customization is an important trend that offers good foundations for achieve the whole system objective or objectives (Lyke et al. 2015).

The customization of RCSs also makes it possible to reach an optimal or desired performance for the controlled and the control system. Thus, **optimization** is the third means to achieve efficiency, effectiveness and reactivity in the RCS. Given the flexibility of the configuration of the RCA and the possibility of customizing the corresponding configuration, the orchestration of the RCS is considered an optimality problem in which the best configuration must be selected according to the system needs. Therefore, the challenge of RCS is to select the optimal configuration to achieve the desired goals. In theory, RCSs can handle the recovery and stability needed to achieve these goals. The literature does address the optimality of the controlled system and the control system approaches (Vlk & Barták 2015; Pach et al. 2014; Zambrano Rey, et al. 2014). However, very little has been written about the optimality of the reconfigurability. In this regard, it is suggested that optimality is a concept related to the controlled system, the control system and the reconfigurable control system. The following illustration explains the need for optimality in the reconfiguration control system.

### **Illustration of the optimality need in reconfiguration control systems.**

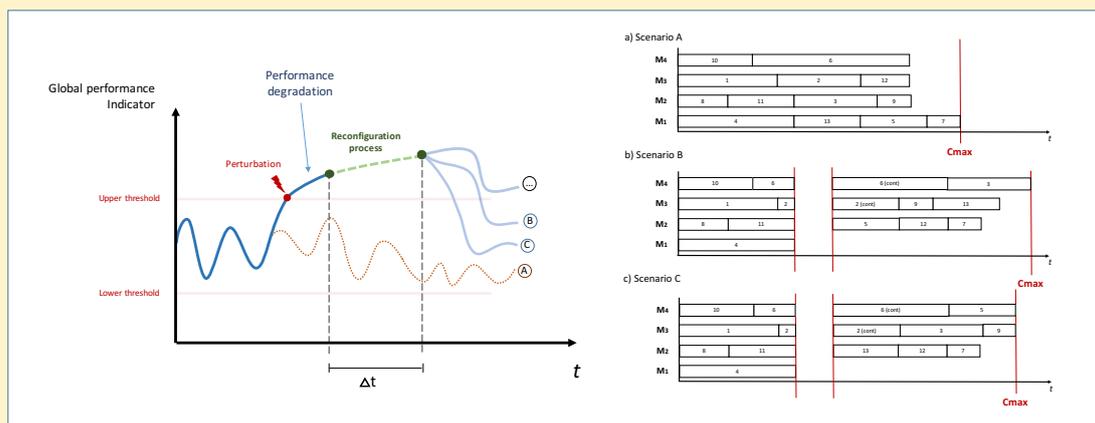
Here we present a case in the manufacturing domain requiring the scheduling and completion of a production order. Scheduling is a decision-making process that aims to allocate resources to several tasks over given time periods to fulfil specific objectives (Pinedo 2012). Therefore,

a pre-emptive parallel-machine scheduling problem is presented in which  $N$  jobs need to be processed in  $M$  machines, and any job can be processed in any machine.

A production order comprised of 13 production jobs  $N = \{1, 2, \dots, 13\}$  is to be processed in a manufacturing facility with four redundant machines  $M = \{M_1, M_2, M_3, M_4\}$ . Each job consists of a single operation; the processing time for each operation is deterministic and known beforehand, the pre-emption of processed jobs is allowed, and it can be processed in any of the available machines. The goal is to minimize the makespan of the production order  $C_{max}$ . In this production problem, a reconfigurable control system (RCS) is implemented for the planning, scheduling and execution of the production order. This includes 14 decisional entities, one decisional entities for each of 13 jobs and one for scheduling coordination of the production order. The initial configuration the control system is a centralized structure where the order starts the execution according to the scheduling proposed by the coordination decisional entity.

For illustration purposes the following three scenarios are shown:

- **Scenario A:** simulates the processing of the production order without any perturbation during execution and no need for reconfiguration.
- **Scenario B:** simulates the processing of the production order with a perturbation (e.g. machine  $M_1$  breakdown) during execution. This scenario maintains a centralized control architecture and the affected jobs are allocated to the available machines (e.g. longest processing time).
- **Scenario C:** simulates the processing of the production order with a perturbation (e.g. machine  $M_1$  breakdown) during execution. This scenario includes a reconfiguration process changing to a fully heterarchical control architecture.



**Figure 8** Illustration of the Reconfiguration processes in reconfigurable control architecture.

Figure 8 shows the performance during the proposed scenarios. These scenarios help to understand the execution of the RCS, the reconfiguration in the control architecture and the

challenges regarding optimality in the reconfiguration process. The challenges are presented based on the timing in the execution: before, during and after the reconfiguration process.

**Before the reconfiguration process:** Execution starts when the scheduled order is launched for production. During execution, the RCS monitors the performance for the global and local indicators and become responsible for detecting any infringement or perturbations of the previously settled upper/lower thresholds. If there are no perturbations, the production performance follows the trajectory of scenario A (See Ⓐ in Figure 8). However, if a perturbation does occur, the system degrades its performance and experiences myopic behaviour. This myopia in the control system context is the inability of the system to acquire and process valid information from its environment in order to determine actions (Zambrano Rey et al. 2013). During this monitoring process, the first challenges regarding the optimization of the reconfiguration process are identified: When should the reconfiguration process start? How should the monitoring process be set in order to detect the perturbation? Should the monitoring process be global or local? These issues are related directly to the optimality of the reconfiguration process under the assumption that a prompt diagnosis and action will contribute to the minimization of the corresponding degradation.

**During the reconfiguration process:** Later in the execution, once the reconfiguration request is acknowledged, the reconfiguration process starts processing in parallel in order to seek out a new configuration for the control system. During this time, while the performance continues to degrade (e.g. no action has been taken), the following challenges can be identified in the optimization of the reconfiguration process: What should the process of reconfiguration be? Which elements, interaction or protocols should be changed in the RCS? How can it be assured that the change elements achieve a better configuration? How much time can be given to the reconfiguration process for its processing? This process may provide different configurations that can definitively guide the control system to different performance trajectories (see Ⓑ and Ⓒ in figure 8). In this sense, the parameterization in terms of scope, techniques and timing will directly impact the performance and optimization of the reconfiguration process.

**After the reconfiguration process:** Finally, when the reconfiguration process has found a new configuration, it must be implemented during execution. For this, the flexible and customizable characteristics of the RCS make it possible to change the configuration in a straightforward way when needed. However, the adoption of the new configuration may need a recovery time, depending on the damage caused by the perturbation. The identified challenge, which is directly connected with the challenges during the reconfiguration process, is the speed of the recovery. A prompt recovery is expected, allowing the adoption of the benefits from the new reconfiguration and therefore enhancing the optimization of the recovery process.

In conclusion, the stability purpose of the RCS is achieved, as the RCA is highly reactive for both the individual components and the structure as a whole, taking into consideration the flexibility and customization. However, the flexibility and customization cannot ensure the selection of the best configuration. The main theoretical implication of this illustration is that the optimality of the reconfigurable mechanism in the RCS is highly sensitive to the

parameterization and the methods used for the reconfigurability. Hence, it is likely that the optimization of the reconfigurable process depends not only on the proper selection of the new configuration, but also on the monitoring, detection, processing and implementation of the reconfigurable mechanism. In this sense, while optimality is needed in the reconfiguration process, it must also be synchronized with the detection and implementation of the reconfiguration.

This framework of reconfigurable control systems defines the key concepts to find, select and evaluate the literature concerning reconfigurable control systems. Since the definition of RCS varies among researchers, however, and the complete nature of RCS remains unclear, this dissertation proposes a definition based on the key concepts of RCS.

A common point in the definitions presented above is that RCSs are based on the principle of exploiting inherent redundancies within system elements and dynamics (Kale & Chipperfield 2005). Nonetheless, these definitions do not agree on the type of reconfigurability and the location where the redundancies should be included. This shows the need for a definition indicating that the RCS is an articulation of the changes of the control architecture that impact the behaviour within the environment. Therefore, a definition that extends the definition proposed by Jiang (1994) is used in this dissertation, as emphasizes the changing of elements, timing and the objective of the reconfigurability: **“An RCS is a control system that is capable of changing its parameters, structures or behaviours automatically on-line to accommodate variations in the process being controlled, maintain as much as possible the efficiency, effectiveness and reactivity of the original system, and specify the guidelines of the control system, such as purpose, policies, strategies and reconfiguration process associated with the control closed-loop”**.

In this definition, it is clear that the key concepts are based on the composition-related and reconfigurability-related features of the RCS. On the one hand, the **composition-related characteristics** focus on the location of the control system and the possible configurations in the structure and behaviour of the control architecture. This is consistent with a characterization of the control systems as it will imply the structural arrangement, which ranges from fully hierarchical, fully heterarchical and semi-heterarchical arrangements, and the behavioural functioning, which includes the actions and scope of the components ranging from simple reflex to decision-making components. On the other hand, the **reconfigurability-related characteristics** focus on the changing mechanism and results with regard to the purpose, strategy and reconfiguration process. Then, in this regard, the key concepts to identify in the literature review are thus the purpose of the reconfigurability, the type of reconfigurability featured and the quality search in the reconfigurability.

### 2.3 Literature review

This section reviews the literature on reconfigurable control systems (RCS) that addresses optimality initiatives and improvements to the efficiency, effectiveness and reactivity of the control system. This review intends to identify the publications with these qualities in order to recognize the

main contributions in this area, construct a characterization model for any RCS, and propose a typology using optimality-based principles. As a result, this review constructs a framework that helps to identify how the inclusion of optimality-based principles impact the optimality and controllability characteristics of RCS and the posterior global-performance enhancement.

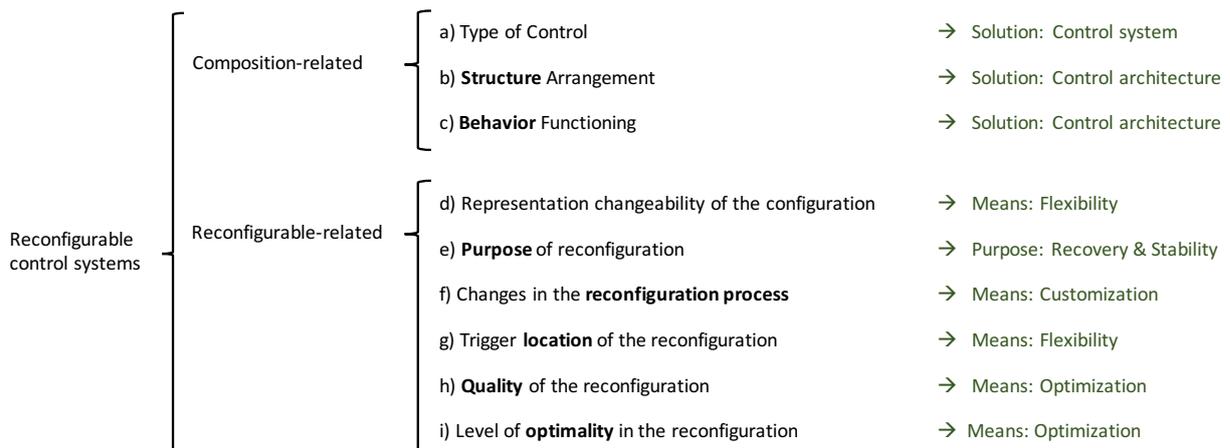
The selection includes contributions of reconfigurable control systems that feature flexibility, customization and optimization of the control architecture with the goal of improving and/or responding to perturbed scenarios. Recent publications on this topic are presented, taking into consideration the functioning of the reconfiguration process, reviewing the representation used in the control architecture, and identifying the optimization-based principles included in the reconfiguration process.

### **2.3.1 Methodology of the literature review**

The review included a search of recent published papers in leading international journals which are indexed in recognized databases. An extensive research of print sources was carried out in engineering and technology databases, including IEEE Xplore, Science Direct, Springerlink, Taylor & Francis, ACM Digital library and Wiley online library. In addition, research was carried out in Google scholar and in the multidisciplinary databases JSTOR and Proquest. Nonetheless, this search included terms of operational research domains in order to aggregate the inclusion of optimality within the reconfiguration. The search included the following keywords: reconfiguration control architectures, types of reconfiguration, reconfiguration flexibility, reconfigurable systems, optimal reconfiguration, reconfiguration optimization, architecture optimization, reconfiguration approaches, adaptable control, adaptive architectures, switching architectures, evolving architectures, reconfiguration states, configuration states, control orchestration, reconfigurable fault-tolerant control and reconfigurable fault-tolerant systems.

After this search, an initial collection of 175 papers relevant to the topic was identified (See appendix C). The titles, abstracts, key concepts and reconfigurability characteristics addressed in the papers were screened in order to assess their relevance to the purpose of this review. This assessment was used as a first filter to narrow the collection down to works with clear explanation of the structure and behaviour of the system, as well as a description of a process that executed reconfiguration from one state to another. A second filter was also applied to indicate the clear inclusion of an optimality feature in the reconfiguration process. As a result of these two filters, a collection of 47 papers were identified for the review in this dissertation. This collection comprises a set of reconfigurable control systems with optimization initiatives in the reconfiguration process. This final collection serves as an input to identify the reconfigurable characteristics of the control architecture, as well as the structure and functioning of this reconfigurable process.

The resulting publications are examined based on composition characteristics and reconfiguration characteristics. The analysis is thus divided in as detailed below. For the purposes of a thorough characterization, each criterion is explained. Figure 9 present the evaluation attributes to assess the reconfigurable control systems in the works found.



**Figure 9** Evaluation attributes of the reviewed reconfigurable control systems

- **Composition-related properties.** This characterization classifies the RCS according to its characteristics in terms of structural arrangement and behavioural functioning during operation. Initially, this classification extracts the configuration of the control architecture without considering the changeability of the reconfigurable system (i.e. configuration under normal conditions). Nevertheless, considering that this classification is for RCSs, it must be taken into account that these characteristics may be changed in the reconfiguration process (i.e. configuration under perturbed conditions). As contextualized in Chapter 1, these contributions are classified according to the following criteria: a) type of control system, b) structural arrangement in the control architecture, and c) the behavioural functioning of the components.

- Type of control system: the contribution is based on how the control operates within the controlled system. IT may be implemented by a central component (centralized class) or by several components (distributed class). In the centralized class, control is based on a unique component that resolves the global control problem. In the distributed class the control system disaggregates the global control problem into sub-problems in order to resolve them collectively between different components. In this classification, the contribution may be either distributed or centralized but not both.
- Structural arrangement in the control architecture: this contribution is classified according to the organization of the components within the control architecture. As detailed in Chapter 1, the organization of components can be classified in fully hierarchical, semi-heterarchical and fully-heterarchical structures (Trentesaux 2009). The classes in the criteria are exhaustive. i.e. they represent all possible types of organization. Note that a system with centralized control is also an organization, but this is not considered in this case as it represents a special case of hierarchical class.
- Behavioural functioning: this is based on the capabilities of the components in the control architecture. Depending on the specification, the behaviour of components can be classified as simple-reflex, goal-based or decision-based (Russell & Norvig 2002). A component with simple-reflex behaviour contains limited processing capabilities within the system (e.g. condition-act rules). A component with goal-based behaviour contains perception and action

capabilities in order to complete a certain desirable situation (i.e. goal), but with limited decision-making capabilities. A component with decision-based behaviour contains both perception/action and decision-making capabilities in order to evaluate and select different alternatives aimed at completion of a certain desirable situation. The exhaustiveness of these criteria is in the capabilities of the component. Considering the perception/action capabilities as a level of knowledge and controllability of the world and the decision-making capability as the level of optimality, the classes are defined from least to the most capable according to these capabilities.

The attributes of the composition-related properties are summarized in figure 10:

<b>Composition-related properties</b>	
<b>a) <u>Type of control</u></b>	
Distributed-based control:	The global problem or decision is divided different components of the control system
Centralized-based control:	The global problem or decision is executed by only one component
<b>b) <u>Structure arrangement</u></b>	
Fully Hierarchical:	All components are interrelated between them with master/slave relationships
Semi-heterarchical:	The components of the system have both master/slave and collaborative relationships
Fully heterarchical:	All components of the system are interrelated with collaborative relationships
<b>c) <u>Behaviour conduct</u></b>	
Simplex reflex:	The components have simple processing capabilities, which with ongoing perception acts with a condition-action rule.
Goal-oriented:	The components have perception/action capabilities and its functioning searches the completion of a certain desirable situation
Decision-oriented:	The components have perception/action capabilities and its functioning execute a decision-making technique to select best alternative towards a desirable situation

**Figure 10** Static, composition-related features of reconfigurable control systems

- **Reconfigurability-related features.** This characterization is created to classify the RCS according to its characteristics with regard to changeability and processes related to the reconfiguration of the control architecture. The results of this classification are particularly important to this review as this will demonstrate how the contributions implement the reconfigurability process within the RCS. Considering the focus of this dissertation, the characteristics explored within the relevant papers correspond to optimality-based principles. In addition, the literature review focusses on the applicability and efficiency of the new configuration state after the reconfiguration process in order to show how the approaches contribute to the controllability and optimality characteristics, respectively. These contributions are classified according to the following characteristics: d) representation of the control solution, e) purpose of reconfiguration, f) type of reconfiguration, g) trigger location of the reconfiguration, h) quality of the reconfiguration, and i) level of optimality in the reconfiguration.

d) Representation of the control solution: the contribution is classified according to how the configuration is represented and changed during the reconfiguration process. In the

characteristic-based class, a specific characteristic represents the control architecture and the reconfiguration process selects the new configuration according an expected result known in advance. In the construction-based class, the reconfiguration process of the control architecture intends to respond to its own needs by repairing the previous configuration and aiming to construct and implement a new one. In this classification, the contribution may be either characteristic-based or construction-based but not both.

- e) Purpose of reconfiguration: the work is categorized according to the intention of the reconfiguration when is implemented. It can be classified as improving performance or responding to a perturbation. In the improving-performance class, the monitoring process executed by the RCS detect that a change of configuration will enhance performance as it contributes positively to the expected results. Conversely, in the responding-to-perturbation class, the monitoring process has detected a deviation between the expected and the real action and in consequence the reconfiguration must be recovered or the degradation of the expected results must be minimized.
- f) Changes in the reconfiguration process: the contribution is classified on the basis of what it is changed in the control architecture during the reconfiguration process. This categorization is based on the approach proposed by J. D. Walsh, Bordeleau and Selic (2007), which classifies changes as structural or behavioural and intra-component or inter-component. The classes are: topology-change, in which a structural inter-component change modifies the organization and relation of components of the control architecture; protocol-change, where behavioural inter-component change modifies the resulted functioning of the components in the control architecture; substitution-change, in which a structural intra-component change modifies a complete component; interface-change, in which a behavioural intra-component change modifies the process inside the component; architectural-change, when an inter-component change modifies both the structure and behaviour of the control system; and internal-change, in which an intra-component change modifies both the structure and behaviour of the control system. It should be remembered that a specific change in the reconfiguration process may lead to another type of change in the long term, but this classification considers only the immediate change in the reconfiguration process.
- g) Trigger location of the reconfiguration: the contribution is classified based on where the perturbation is detected within the control architecture. The reconfiguration is triggered in a global component that monitors the global metrics (i.e. global triggering) or in a local component that monitors a local metric (i.e. local triggering). This is done either to improve the performance or respond to a perturbation. However, in some cases the contribution may consider both classes for triggering the reconfiguration process.
- h) Quality of the reconfiguration: the work is classified based on how the new configuration is evaluated, either in terms of the possible implementation or assessing the quality achieved in the reconfiguration process. The reconfiguration may have a function or indicator that evaluates the efficiency and/or effectiveness of the reconfiguration. On the one hand, in what we call the performance-based class, the reconfiguration assesses possible configurations or the new configuration implemented by evaluating the process with a fitness or quality function. On the other hand, in the feasibility-based class, the reconfiguration assesses possible

configurations or the new configuration by evaluating the changeability of reaching a feasible new configuration to continue with execution.

- i) Level of optimality in the reconfiguration: the contribution is classified on the basis of the intended level of optimization in the reconfiguration process. This categorization uses the framework for scheduling methods proposed in Baker (1998), where the process is classified according to the approximation to the optimal solution. The class is defined according to the optimality of the reconfiguration process: optimal (e.g. mathematical programming, combinatorial optimisation), near-optimal (e.g. approximation algorithms, probabilistic algorithms), towards-optimal (e.g. neural networks, evolutionary algorithms, swarm algorithms) and heuristic (e.g. artificial intelligence, priority rules, iterative simulation). An additional optimality class called reaction transition is created to classify the approaches with no specific optimality intention (lack of optimality) and implemented just for the purpose of continuing the execution.

The attributes of the reconfiguration-related properties are summarized in figure 11:

### Reconfiguration-related properties

#### d) Representation changeability

- Characterized-based: A specific characteristic represents the control architecture and the reconfiguration process selects the new configuration according an expected result acknowledged beforehand
- Constructive-based: The reconfiguration process of the control architecture intends to respond to its own necessities by repairing the previous configuration and aiming to construct and implement a new one

#### e) Purpose of reconfiguration

- Improving performance: It intends to improve the global performance of the system either for the physical or the control performance indicators
- Responding to perturbation: It intends to respond and adapt to the uncertain events occurred in the physical or control system

#### f) Change types of reconfiguration

##### Structure:

- Topology-change: This type involves the addition and/or removal of components, changes the structure and modifies the communication within the system
- Substitution-change: This type involves substituting one component (module, variable, etc.) for another and changes the interaction featured of this component

##### Behaviour:

- Protocol-change: This type involves changing the behaviour result of a component with other characteristics and changes the influence of this component within the system
- Interface-change: This type involves changing the decision process inside the component and changes the behaviour of this component

##### Structure-Behaviour:

- Architectural-change: This type changes global and local properties in the structural and behavioural characteristics (Topology and Protocol together)
- Internal-change: This type involves changing the implementation of the global or local properties and changes the operations and state (Substitution and interface)

#### g) Trigger location

- Global level triggering: The reconfiguration process is started by any global component regarding the detection of an issue (Improvement or perturbation case) from a global indicator
- Local level triggering: The reconfiguration process is started by any local component regarding the detection of an issue (Improvement or perturbation case) from its own execution

#### h) Quality of the reconfiguration

- Performance-based: The reconfiguration evaluates the potential configuration or configurations with a quality function
- Feasibility-based: The reconfiguration searches a new configuration that is feasible and permits the continuity of execution

#### i) Level of optimality

- Optimal level: Reconfiguration is optimized searching the best alternative from the problem, e.g. exact methods, as mathematical programming and combinatorial techniques
- Near-optimal level: Reconfiguration searches a close the solution to the optimal solution, e.g. approximation algorithms, goal programming and probabilistic algorithms
- Towards-optimal level: Reconfiguration attempt an optimal solution but not guaranteeing optimal or near-optimal, e.g. artificial intelligence, neural networks, evolutionary algorithms and swarm algorithms
- Heuristics level: Reconfiguration uses techniques that seem a good performance but which cannot be proven to be optimal or even evolving toward an optimal solution, e.g. priority rules, iterative algorithms and dedicated techniques
- Reaction transition: It does not contain a specified technique with optimality features.

Figure 11 Dynamic features of reconfigurable control systems

### 2.3.2 Characterization and analysis of reconfigurable control systems

Table 2 presents the characterization of the 47 papers addressed and evaluated in this literature review. Appendix D presents a short description of the aim and proposal of these papers.

## Chapter 2. Reconfigurable control systems: literature review

**Table 2** Characterization of selected papers from the literature review of reconfiguration control systems

	Composition properties			Reconfiguration properties											General description	
	a	b	c	d	e	f			g		h		i		j	k
a. Type of control	Distributed based control Centralized based control	Fully hierarchical	Semi-Heterarchical Fully heterarchical	Goal-oriented behaviour Decision-oriented behaviour Not specified	Characterized-based representation Constructive-based representation Improving performance Responding to perturbations Both	Topology-change Protocol-change	Substitution-change Interface-change	Architectural-change Internal-change	Global Triggering Local Triggering Both	Not specified	Performance-based Feasibility-based Not specified	Not used	Optimal Near-Optimal Toward-Optimal	Heuristic Reaction transition Not Specified	Domain	Technology
b. Structure arrangement																
Srini and Shriver. (1984)	•		•		•			•	•					•	Airport Logistics	Automation
Pete et al. (1995)	•	•		•	•						•			•	Administration	Not technological
Balasubramanian et al. (2001)	•		•	•	•		•		•		•			•	Manufacturing	Automation
Wills et al. (2001)	•			•	•		•		•		•			•	Engineering	Cyber Physical Systems
Brennan et al. (2002)	•	•		•	•			•			•			•	Manufacturing	Automation
Eisenring and Platzner. (2002)	•	•	•		•	•			•		•			•	Electronics	Artificial Intelligence
Pellegrini and Riveill. (2003)	•		•		•		•				•			•	Computer science	Artificial Intelligence
Rawashdeh and Lump. (2006)	•	•		•	•			•	•		•		•		Engineering	Artificial Intelligence
Leitao and Restivo (2006)	•	•		•	•	•			•		•		•		Manufacturing	Cyber Physical Systems
Bossuet et al. (2007)	•	•	•		•	•		•			•		•		Electronics	Artificial Intelligence
Cao et al. (2007)	•	•	•		•	•	•		•		•		•		Robotics	Artificial Intelligence
Chunjie et al. (2007)	•	•	•		•	•				•			•		Informatics	Artificial Intelligence
Patouni et al. (2007)	•	•		•	•	•			•		•		•		Communications	Artificial Intelligence
Youssef et ElMaraghy. (2007)	•	•	•		•		•		•		•		•		Manufacturing	Cyber Physical Systems
Yu, Zhang, and Klemm (2007)	•	•		•	•		•		•		•		•		Manufacturing	Artificial Intelligence
Gumzej et al. (2009)	•	•		•	•	•			•		•			•	Computer Science	Cyber Physical Systems
Rooker et al. (2009)	•	•	•		•	•	•		•		•			•	Manufacturing	Cyber Physical Systems
Holvoet, Weyns and Valckenaers (2009)	•	•	•		•	•			•		•		•		Manufacturing	Automation
Hsieh et al. (2010)	•	•	•		•		•		•		•			•	Manufacturing	Automation
Renna. (2010)	•	•		•	•		•		•		•		•		Manufacturing	Artificial Intelligence
Valente et al. (2012)	•	•	•		•	•			•		•		•		Manufacturing	Artificial Intelligence
Raileanu et al. (2012)	•	•		•	•	•			•		•			•	Manufacturing	Automation
Eddahech et al. (2013)	•		•		•	•		•		•			•		Computer Science	Automation
Oh et al. (2013)	•	•	•		•	•		•	•		•		•		Supply Chain	Artificial Intelligence
Ostroumov et al. (2013)	•	•	•		•	•		•	•		•			•	Computer Science	Automation



In general, the literature review shows that some contributions address the optimality requirement within the reconfiguration process. However, these contributions do not have a clear systematic process that addresses the optimality-based principles. The level of optimality shows that reconfigurable systems do not methodically consider an optimal mechanism within the reconfiguration purposes. This tendency was expected, as the optimality techniques require a large amount of computational time during execution (Kumar et al. 2008). Nevertheless, the inclusion of reconfiguration processes with heuristic, towards-optimal and near-optimal levels of optimality confirm that the researchers seek not only simple feasibility objectives representative of effectiveness goals, but also other types such as optimality of reconfiguration, representative of efficiency goals.

More specifically, most of the approaches in this review concentrate on reaction transition and heuristics. In the reaction-transition type, the control system performs a reconfiguration prioritizing the continuity of execution regardless of the search for an optimal configuration. For example, Sрни and Shriver (1984) propose a methodology to manage the communication system in airports based on the principles of dataflow for reconfiguration purposes. The system contains communication switches that turn on/off based on patterns from condition-action thresholds. The system arrives at a new configuration by changing certain attributes from the system and evaluating the feasibility of this new mode. However, this case does not have an iterative mechanism or comparison of alternatives that searches for a better configuration. Other examples of reaction transition include Liu et al. (2016), Xia et al. (2016), and Eisenring and Platzner (2002). In the heuristic type, Dennis et al. (2014) propose a hybrid agent architecture for controlling an autonomous system based on a new configuration that is viable. The reconfiguration process in this approach changes the behaviour of agents according to pre-defined states. These states provide different alternatives to evaluate during the reconfiguration process and ensure that the system will resume the execution with a new agent behaviour. However, this reconfiguration features a limited number of alternatives and total completion is not assured. Further approaches of the heuristic kind are Zambrano Rey et al. (2014a), Da-Silva, Blos and Junqueira (2014), Valente et al. (2012) and Brennan, Zhang and Xu (2002).

Some approaches contribute to implementing towards- or near-optimal techniques within the reconfigurable control systems. Considering the limited time of the reconfiguration process, these approaches execute an iterative algorithm to evaluate and select the best configuration from among different candidates. An example of this approach is found in Novas et al. (2013). The authors propose a holonic framework that integrates a centralized and a distributed system to minimize the gap between the scheduling and the operation execution via delegate MAS and PROSA. The reconfiguration process, inspired by exploratory ants from swarm optimization algorithms, evaluates different solutions by exploring several scenarios that predict the course of action of the holons. In consequence, the holon can choose the best course of action based on the information of the exploratory solution and the current state of the physical system. In another example, Chunjie et al. (2007) propose the use of a genetic algorithm to dynamically reconfigure the layout of a networked control system. The reconfiguration process takes a representation of the architecture (i.e. the routing path and encoding scheme) and changes its representation with the objective of minimizing the network-time delay. Both examples seek out a new configuration based on optimality algorithms. Unfortunately, these approaches are limited as they present only an approximation of the optimal solution. Nonetheless, this solution responds, in feasibility and performance improvements, to possible perturbations in the physical environment. Other examples include Dias-Ferreira et al. (2016), Michalos et al. (2015), Barbosa et al. (2015) and Youssef and ElMaraghy (2007).

For the reconfiguration process, the papers can be examined according to two criteria: the *characteristics of the reconfiguration technique* and the *representation used during this reconfiguration*. For the characteristics of the reconfiguration technique, the reviewed works can be divided into two clusters: reconfiguration as an emergence functioning from a collective behaviour and reconfiguration as a change of certain components of the system (e.g. components, modules, algorithms, decision-making techniques, variables, parameters, and many others). For representation used during reconfiguration, the reviewed works can be classified based on the evaluated alternatives during reconfiguration. In this regard, two categories characterize the reviewed approaches: the evaluation of several candidates from set of possible configurations (i.e. population-based techniques), or evaluation of a configuration by changing part or parts of the current configuration (i.e. trajectory-based techniques).

**Characteristics of the reconfiguration technique:** As we have seen, the exchange or swapping of a part of the system is a reconfiguration technique observed in some of the reviewed papers. In this case, the reconfiguration process evaluates an alternative or alternatives (population or trajectory-based) and changes a specific part or parts of the system to bring about a change in behaviour. The advantage of this technique, which clearly needs an evaluation function, is that it completely changes the control strategy of the system with a new configuration, with its own objective, control strategy and system capabilities. This technique aims to compensate for perturbation damage with a new configuration that absorbs the degradation. Therefore, it intends to mitigate the degradation and/or recover the final output of the system. However, it has certain disadvantages. First, the recovery process and the evaluation of the alternatives may require time to calculate a best or satisfactory new configuration. This is not always possible due to the limited time of reconfiguration. Second, considering that a reconfiguration period is required, the system needs a default configuration to execute while the new configuration is synchronized. This configuration must reactively execute the system and may intend to optimize its operations. Third, synchronization is a difficult task. Considering that the evaluation of new alternatives is based on a specific current state of the system, at the moment when it is synchronized the system might have a different state which may only partially adopt the new configuration. In this sense, a synchronization technique is needed which can implement the new configuration. The exchange or swapping of a part of the system is a technique that promises to compensate for the degradation caused by perturbation, but certainly it implies the inclusion of additional methods to ensure the continuity of execution.

**Representation used in the reconfiguration:** Another approach in the reviewed papers is an emergence functioning from a collective behaviour. The reconfiguration process in these cases is not an intrusive change in the system, but rather a gradual accommodation of the behaviour according to the interaction between the components (either in the population or trajectory-based). The advantages of this technique are that the system is highly reactive to any perturbation. It can consequently sustain the execution of the physical system and it features a highly modular and exchangeable system (with no need for a default mode). In addition to these advantages, with regard to the reconfiguration process, the monitoring process is distributed, the changes in the behaviour of the components do not intervene in the operation of other components and the synchronization of the new configuration is immediate within the execution. The disadvantages are that the emergence expected after the reconfiguration is not guaranteed to converge to an efficient global performance; only the feasibility of the new configuration is assured. In addition, although the synchronization is immediate after the reconfiguration process, the time required for the system to recover from the perturbation is uncertain.

## Chapter 2. Reconfigurable control systems: literature review

In fact, it may only minimize the degradation caused. Emergence from a collective behaviour is a good technique to assure continuity and respond to perturbations. However, it may be used to minimize degradation rather to find a configuration for optimal recovery. Table 3 illustrates the classification of these approaches according the described criteria.

**Table 3** Typology of the reviewed literature according to the reconfiguration technique.

<b>Population-based techniques</b>			
	<b>Emergence for collective behaviour</b>	<b>Change or swap of components</b>	
	Holvoet, Weyns and Valckenaers (2009) Hsieh et al. (2010) Ostroumov et al. (2013) Sedighzadeh et al. (2014)	Wills et al. (2001) Rawashdeh and Lumpp (2006) Cao et al. (2007) Chunjie et al. (2007) Gumzej et al. (2009) Renna (2010) Raileanu et al. (2012) Oh et al. (2013)	Dennis et al. (2014) Indriago et al. (2015) Liu et al. (2015) Oliveira and Barbosa (2015) Borangiu et al (2015) Jimenez et al. (2015a) Gerostathopoulos et al. (2016)
<b>Trajectory-based techniques</b>			
	<b>Emergence for collective behaviour</b>	<b>Change or swap of components</b>	
	Srini and Shriver (1984) Pete et al. (1995) Pellegrini and Riveill (2003) Leitao and Restivo (2006) Bossuet et al. (2007) Patouni et al. (2007) Yu, Zhang and Klemm (2007)	Da Silva et al. (2014) Streit et al. (2014) Barbosa et al. (2015) Leitao and Barbosa (2016) Vasconcelos et al. (2016) Xia et al. (2016)	Balasubramanian et al. (2001) Youssef et ElMaraghy (2007) Rooker et al. (2009) Valente et al. (2012) Pach et al. (2014) Zambrano Rey et al. (2014a) Zielinsly et al. (2015) Michalos et al. (2015) Gorecky et al. (2016)

In addition to the characteristics analysed above, it was observed that some contributions execute a process parallel to the execution in order to assist the reconfiguration process when needed. The reconfiguration process has a module or component that calculates an alternative configuration in parallel regardless of whether or not there is a perturbation. This periodic and parallel evaluation (population or trajectory-based) makes it possible to minimize the time for evaluating new configuration alternatives and offers constantly a new candidate ready to be used when needed. The advantages in this case are that there is always an adequate alternative configuration to implement that considers a more recent online situation than the last one retrieved. However, the disadvantages are that this requires extra resources for the parallel calculation, a robust synchronous messaging is needed and, although the configuration was evaluated based on a very close current state of the system, a synchronization technique may still be needed. This is a new technique used in reconfiguration control system. It is worth exploring this technique further because it resolves many drawbacks of previous techniques and promises improved responsiveness from the reconfiguration process. This reconfiguration technique approach is proposed for both population-based (Eddahech, Chtourou and Chtourou 2013; R. W. Brennan, Fletcher and Norrie 2002) and trajectory-based (Brusafferri et al. 2014; Dias-Ferreira et al. 2014; Novas et al. 2013) techniques.

### 2.4. Summary

The main finding of this literature review is that reconfigurable control systems are still a work in progress and a topic in development. The reviewed literature shows that these approaches are highly dedicated to a specific domain and do not contribute to constructing a general concept. Therefore, the literature review of this dissertation suggests that, considering the need for a system that contributes the resolution of society's complex and dynamic demands through technological advancement, there remains a need for a generic framework of reconfigurable control systems that can be implemented in any domain. This implies that the design method and the architecture should both be proposed.

More specifically, and targeted to the objectives of this dissertation, the following conclusions may be drawn from the literature review:

a) Reconfigurability control systems are a promising approach for managing both the improvement of system performance and sudden changes in the controlled system and control system. As seen in this chapter, researchers can use RCS as a solution to meet the needs for monitoring, managing and controlling the system to meet sustainability requirements (e.g. controllability, optimal performance, adaptability and reactivity). This approach enhances the capabilities of the control system by providing a tailor-made control solution to compensate for various degradations caused by perturbations.

b) The implementation of the required technological solutions requires development of a highly flexible control architecture to provide the combinatorial alternatives to the reconfiguration mechanism. In this sense, reconfigurable control systems serve as a mechanism to provide a customizable flexibility that adapts properly to the system online requirements.

c) The reconfiguration process must adhere optimality-based principles to ensure the adoption of the best possible control solution subjected to the reconfiguration restrictions. The literature review shows some advances in the inclusion of optimality features in reconfiguration mechanisms. The inclusion of this principle might jeopardize the reactivity and fault-tolerant characteristics needed in the control system. However, the reconfiguration process is a trade-off between optimality and reactivity that can be customized to meet the system needs. The challenge is to provide a reconfiguration technique that finds the proper balance in this trade-off.

d) Last but not least, the reconfigurable mechanism must define organized methodologies to answer the following questions. *When*: it must be known the best time in the execution to perform a reconfiguration. *What*: it must be known exactly what should be changed in the control architecture to achieve recovery during the reconfiguration. And *How*: it must be known how the identified component(s) should be changed in order to reach a new and improved configuration in terms of the control solution.

Considering these conclusions, the requirements for a general RCS are as follows:

1. A reconfigurable control architecture must be constructed with the ability to change the structure and behaviour online in order to adopt a control solution able to respond to perturbations and minimize the degradation in efficiency, effectiveness and reactivity.

2. A mechanism must be included which, with a reconfiguration process, guarantees the continuity in the execution of the controlled and control systems. Therefore, systems require redundant devices, processes and/or protocols at specific levels to ensure continuity.
3. A flexible control architecture is needed that provides different solutions for the global control problem. The reconfigurable mechanism will then use this flexibility to evaluate different control objectives to achieve certain effectiveness and efficiency objectives.
4. Several parallel processes must be implemented to provide synchronous instructions in the control system architecture. This may involve parallel processes to evaluate configuration alternative solutions within the system execution, but certainly this requires computation time for each computational device.
5. Given that the reconfigurability may be applied in several domains, it is likely that the reconfiguration monitoring and action might need a standardized communication language or interfaces for interoperability communication, either for physical devices or software applications.
6. The orchestration of predictive and reactive decision-making techniques must feature an optimality and reactivity control solution for the controlled system.
7. Considering that the objectives of a control system and controlled system may search for a desired performance in the result and during the execution, it must be taken into account that the reconfiguration control system needs to balance a trade-off between efficient and effective performance to achieve sustainable development of system execution.

The next chapter describes POLLUX, a proposed reconfigurable control architecture featuring improved controllability and optimality characteristics in the reconfigurable process.

# Chapter 3

## Pollux: a reconfigurable control system

### 3.1 Introduction

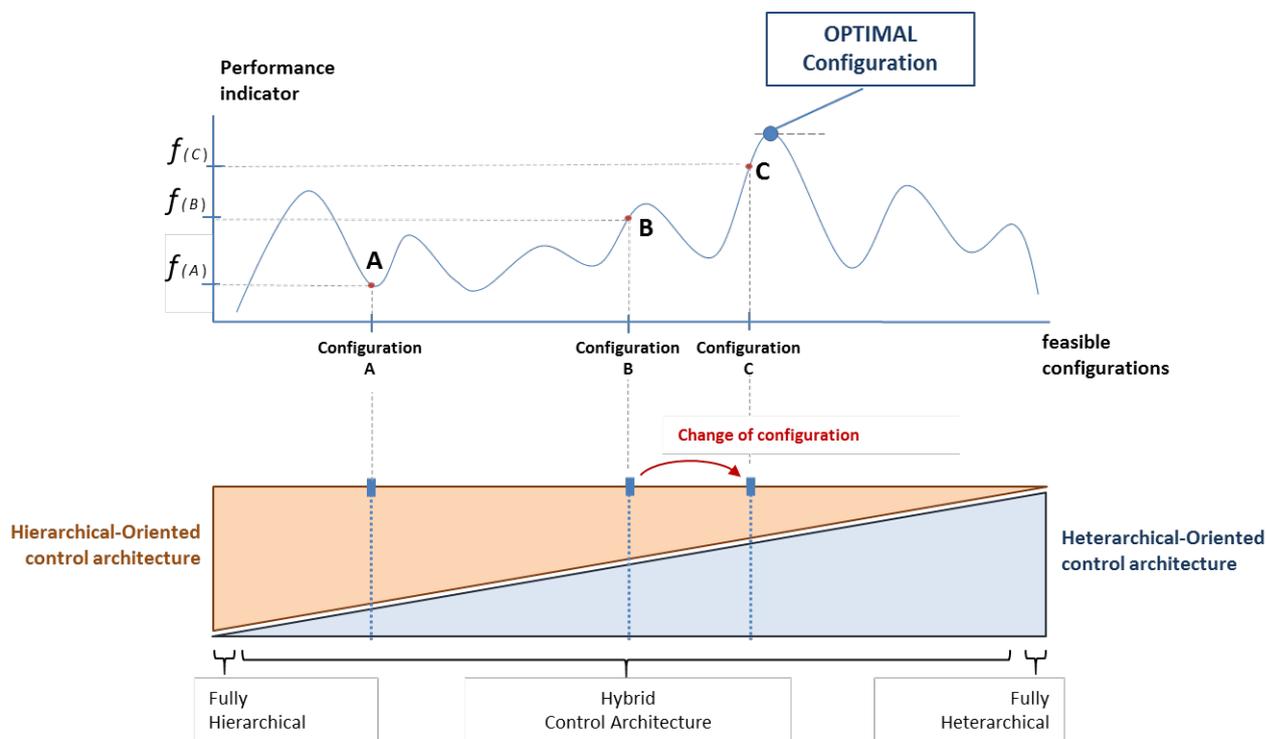
The previous chapter presented a literature review regarding reconfigurable control systems and presented the key concepts for designing a reconfigurable control system with optimization initiatives. Considering the contextualization of Chapter 1 and the analysis of the theoretical framework in Chapter 2, a reconfigurable control system for discrete-event systems is proposed in this chapter. This chapter aims to describe in detail the composition, dynamics and reconfiguration behind the reconfigurable control architecture of Pollux.

Section 2 of this chapter starts by describing the specifications of Pollux, which is based on the concept of executing a change of the control architecture to attain optimal configuration, and the highlighted requirements of reconfigurable control systems as described in Chapter 2. The proposed reconfigurable control system is detailed in section 3. This description is divided into three subsections: a general description of Pollux with the main characteristics and the objectives of this approach (subsection 3.1), a description of the control architecture (subsection 3.2), and a description of the reconfiguration mechanism (subsection 3.3). Section 4 extends the explanation of the reconfiguration technique module (as part of the reconfigurable mechanism) and deepens in the improving optimality and controllability. Finally, a summary of the attributes and features of Pollux is presented in section 5.

### 3.2 Specifications of the Pollux approach

Pollux is a reconfigurable control system designed to manage and adjust the architecture of a control system optimally and in real-time, either to guide operational execution or as a response to unexpected changes in a controlled system. This RCS is a reconfigurable system designed specifically

for discrete-event systems. It contains a three-stage reconfiguration mechanism that detects a reconfiguration intention, executes a reconfiguration technique, and modularly implements a new architecture configuration to adjust and improve system performance. To this end, Pollux executes a reconfiguration process that searches for a more appropriate configuration according to the current needs and conditions of the system. The design of Pollux is based on the concept of executing the change of the control architecture to attain the optimal configuration (Jimenez et al. 2015a).



**Figure 12** Changing the configuration of the control architecture to attain optimal performance

Figure 12 illustrates attainment of the optimal performance indicator by changing the configuration of the control architecture. This process starts by assuming that a control system is built with a flexible and changeable configuration capable of adopting different feasible configurations depending on the degree of influence between the hierarchical and heterarchical structure-orientation. At the same time, each of these feasible configurations is assumed to supply a different emergence behaviour to the control system. In consequence, the system can find an optimal configuration from among the feasible configurations.

To achieve this, Pollux consists of a control architecture with flexible characteristics that can be changed (at the component, structure and behaviour levels) during real-time execution. Pollux can be implemented in the operations of any domain. It serves as a reference architecture for a control system that both seeks out reconfigurability features and improves the optimality and controllability in this reconfigurability. Thus, the main specifications of Pollux are as follows:

**Spec 1.** Responds to the distributed allocation of the global control problem according to the participants in the control system. Pollux divides the global control problem into simple sub-problems that work together to meet the objectives of the control system (including the controlled system).

- Spec 2.** Composed of decisional entities, i.e. intelligent and autonomous entities that use a decision-making technique to search for the best behaviour alternative to reach their own objective or objectives. Pollux uses customizable decisional entities that represent the components of a controlled system that, either for global coordination tasks or local execution and negotiation processes, take responsibility for the completion of the assigned sub-problems. These decisional entities can then support the predictive and reactive decision-making techniques needed to provide efficient, effective and reactive performance in the execution of the control architecture.
- Spec 3.** Constructs the control architecture in two layers to allocate the components according to the responsibilities and contributions needed in the control system. These entities may be allocated in the coordination layer if these contribute to the coordination of tasks and organization of the decisional entities, or the operational later if they contribute to the execution of the control system and the completion of the assigned sub-problems.
- Spec 4.** Considers a flexible and changeable constitution of the decisional entities that permits these features to exhibit flexible behaviours during execution. Pollux selects a sub-set of parameters within its constitution (i.e. governance parameters) that can be changed repeatedly to set the rules of conduct for each decisional entity. Although in general terms this flexibility makes it possible to choose any parameter from the decisional entity (i.e. objective to follow, decision-making technique, evaluation criteria, etc.), this dissertation focusses on the flexibility given in the interactions between the different components (i.e. type of relationships).
- Spec 5.** Introduces an operating mode as a representation of the configuration of the control architecture, whose composition represents a control solution with its own objective and strategy to control the controlled system. Pollux gathers the governance parameters to compose the operating mode. This operating mode provides various alternatives to the control problem and represents an approach to identify and evaluate a specific alternative before implementation.
- Spec 6.** Includes a specific reconfiguration mechanism that follows an iterative improvement search process to improve reactivity and minimize degradation in global performance. Pollux divides the reconfiguration mechanism onto three modules to solve the main challenges of RCS.
- Spec 7.** Builds a reconfiguration mechanism based on the use of a reconfiguration-technique module with optimization-based principles. Pollux creates a reference reconfiguration mechanism that can host any optimization process, such as metaheuristics, neural networks, evolutionary algorithms or heuristics. Pollux uses these techniques in order to find an adequate operating mode according to the reconfiguration objectives. Pollux includes this module to respond to the “how to reconfigure” question.
- Spec 8.** Includes a reconfiguration-triggering module responsible for activating the reconfiguration mechanism when it detects a degradation of the system either on the global or local level. Pollux includes this module to respond to the “when to reconfigure” question.
- Spec 9.** Includes a reconfiguration synchronization module that implements the new operating mode and performs the necessary changes for implementation. Pollux includes this module to support the “how to reconfigure” question.

### 3.3 Pollux: a reconfigurable control system

#### 3.3.1 Description of the proposed approach

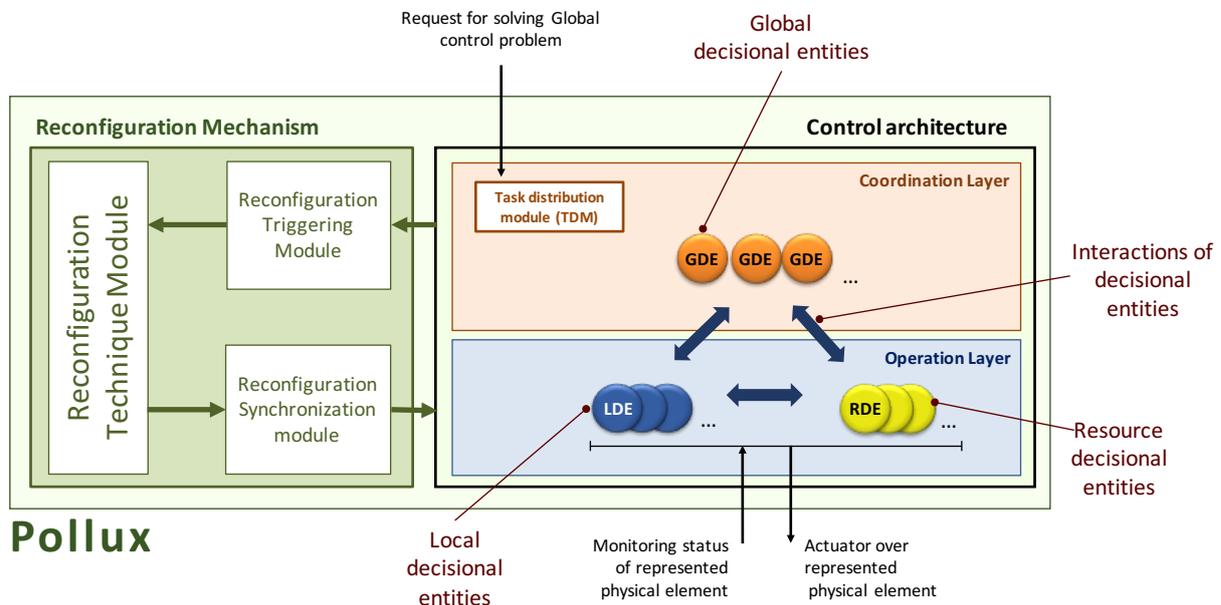


Figure 13 General illustration of the reconfigurable control architecture of Pollux

Pollux is a reconfigurable control system to be used for the automatic control of a discrete-event systems, such as manufacturing execution system (MES) or delivery systems (DES). During implementation, Pollux takes the objectives of the control system and the requirements of the control system to resolve a global control problem. In this sense, Pollux is able to resolve **a global control problem assigned through the control system.**

Pollux is comprised of two subsystems (See figure 13):

- **Reconfigurable control architecture:** the reconfigurable control architecture (hereinafter, control architecture) is a dynamic hybrid architecture that controls the controlled system in a combined hierarchical and heterarchical manner. This architecture consists of decisional entities (i.e. global, local and resource decisional entities) connected to interact and participate in a collective process to meet global and local objectives.
- **Reconfigurable mechanism:** the reconfigurable mechanism is a subsystem that governs the flexible and changeable capabilities of the control architecture to improve the optimality and controllability of the control system.

The next sections describe the procedures and methods used by the Pollux architecture.

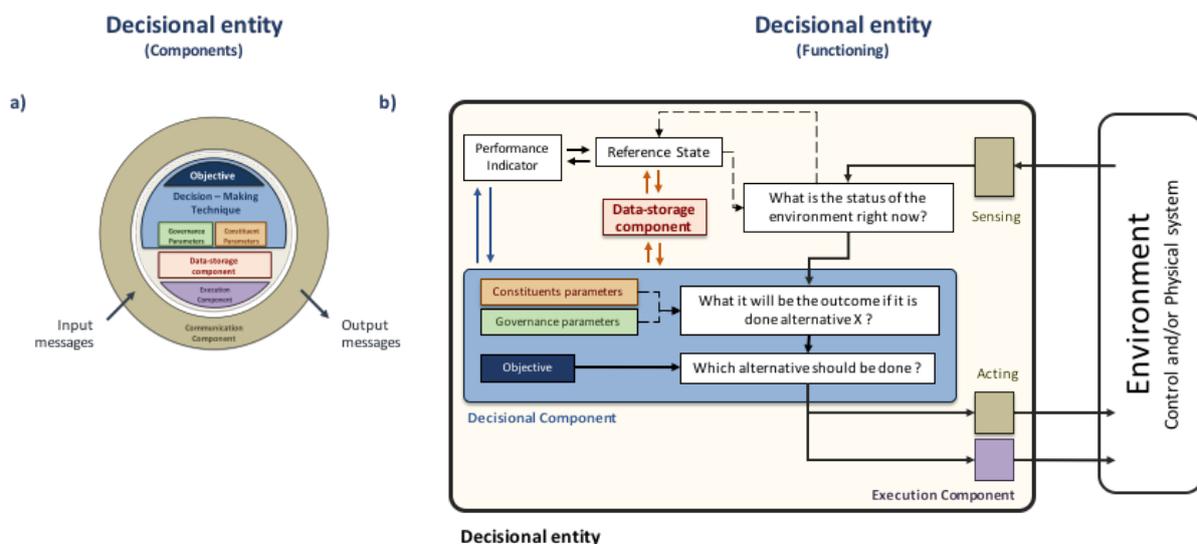
#### 3.3.2 Control architecture

The control architecture of Pollux is an organized set of decisional entities, each of which is responsible for completing a specific task allocated from the distribution of the global control problem (**Spec 1 & Spec 2**). This subsection defines the general composition and functioning of the decisional entities and describes the three types of decisional entities that comprise the control architecture.

### 3.3.2.1 Decisional entity

As indicated in the specifications (**Spec 2 & Spec 4**), Pollux is comprised of decisional entities that act together to achieve a specific purpose defined by the global control problem. A decisional entity has reactive and cooperative capabilities to achieve its designated purpose (named default behaviour of decisional entity). The decisional entity is based on the goal-based agent (Russel & Norvig, 2002) that evaluates the corresponding course of action through a decision-making technique. The goal of the decisional entity is to achieve the desirable state of the entity, either based on a specific metric (e.g. nominal value, percentage value, interval range or limited threshold, etc.) or characteristic (e.g. geographical position, completion state, destination arrival, etc.). If an objective requires a completion of a sequence of actions, each action turns into a goal and the decisional entity successively completes each action until the entire sequence is completed.

A decisional entity which runs a repetitive closed-loop process is capable of sensing, processing, storing and acting through the control system environment. Each decisional entity is made up of decisional, communication, data-storage and execution components. The decisional component, as the core element, is a processing unit that manages the behaviour and actions of the entity based on a decision-making technique. The decisional component contains the objective(s) assigned to the entity, the corresponding decision-making technique, the constituent parameters and the governance parameters. The communication component actuates the data transmitter within the control/physical system and other entities. The data-storage component is responsible for consolidating the current and historical information of the entity. Finally, the execution component is responsible for initiating the actions of the entity. For an illustration of this structure, see Figure 14a.



**Figure 14** Composition and functioning of the decisional entity

For Pollux, the main characteristic of the decisional entities is that the decisional process is directed by the governance parameters. These are an explicit set of parameters (defined at the beginning but with the possibility of being changed during execution) that define the attributes and rules of conduct that dictate the action of the decisional entity. For example, the governance parameters can define the objective(s), the interrelation with other entities, the decision-making technique, the roles of the entities in the shared environment, the priority of objectives in a multi-objective environment, or any other attribute that defines the behaviour of the decisional entity. The process of the decisional entity starts by sensing the current state through the communication component (See figure 14b). Then, in order to accomplish the objective(s), the decision-making technique is activated based on the current entity configuration. The decision-making technique evaluates different action alternatives, chooses an action solution based on its own objective or reference state, and implements the actions through the execution component. The decisional entity is parameterized with a decision-making technique with any type of model or method that acknowledges the alternatives, evaluates these alternatives and selects a solution as the course of action of the decisional entity. Some examples of this decision-making technique are simple reflex algorithms, priority rules, dedicated heuristics, metaheuristics and exact methods. Therefore, these are able to hold any predictive or reactive technique to dictate the decisional entity behaviour. The configuration of the decisional entity, including the governance parameters and the decision-making technique, is a key driver in the flexibility and capability achieved in the proposed approach.

### 3.3.2.2. Types of decisional entities

There are three types of decisional entities that comprise the control architecture: local decisional entities (LDE), resource decisional entities (RDE) and global decisional entities (GDE). This division of components of the control architecture responds to the distributed allocation of the global control problem among the participants in the control system. This division is complemented from the concept of decisional entities proposed by Zambrano Rey et al. (2014a).

**Local decisional entities (LDEs):** The LDEs, as an instantiation of the decisional entity, represent the entity that have an activity to be completed, and the objective is to complete this activity (divided sub-problem of the global control problem). Examples of LDEs in the manufacturing and health-care domains include products requested in a manufacturing cell or patients requiring a specific medical procedure, respectively. This type of entity is not only responsible for resolving a pre-assigned and distributed control problem, but also manages the online (real-time) decisions to meet the assigned objective. Each LDE has information concerning the achievement of the assigned control task, such the list of steps for achieving the objectives and information regarding the environment (local information). The performance indicator of the LDEs is a metric that measures completion of the assigned objectives and serves as a reference state for the function of the entity. In Pollux, the LDEs have two different and mutually exclusive types of behaviours included in the decision-making technique. On the one hand, in a more **autonomous behaviour** and considering that the LDEs are responsible for the adaptation of the control system, the decision-making technique included in the LDE is a reactive technique that responds during the online decision. These reactive techniques, which is mostly used for heterarchical relations, may be simple reflex, collaborative decision, priority rules or heuristic, and within a limited amount of time they achieve continuous operation during execution. On the other hand, in a more **submissive behaviour**, the decision-making technique included in the LDE is a set of instructions given by another entity as an action derived from of a hierarchical relation. The type of behaviour exhibited by the LDEs at a certain time of the execution is indicated either by

the governance parameters or by an immediate change in order to respond to an unexpected event (improvement or perturbation). Further information regarding this change function is provided in the description of the reconfiguration mechanism.

**Resource decisional entities (RDEs):** The RDEs manage the service-orientated resources located in the controlled system (machines, medical operations, airport check desk, etc.) and are responsible for carrying out the request made by the LDEs and supporting the controlled system in fulfilling the objective assigned in the global and divided control problem. These entities have all the information related to the resource, including operation times, localization, capacity information and other detailed information. The objective of the RDEs is to carry out requests from the LDEs while maintaining the resource-related objectives given by the global control problem. The decision-making technique of the RDEs is mainly simplex reflex, based largely on implementation of the task requested by the LDEs. The decision-making technique may include an improved method with collaboration decisions, priority rules or heuristics due to the need to optimise the resource behaviour/use during execution. Therefore, the main performance indicator of the RDEs is the effectiveness and efficiency in the completion of the LDE tasks. However, both the decision-making technique and the performance indicator may also include reactive indicators.

**Global decisional entities (GDEs):** The GDEs are specific entities that collaborate in the completion of the global control problem by supporting coordination of activities and ensuring the achievement of the global objectives. Each GDE has information concerning the completion of the global problem and contains a specific objective to carry out this completion. If the global control problem contains more than one objective, the problem might be partitioned to allocate the same problem with different objectives to several GDEs, or it might be allocated to a unique GDE but configuring its own decision-making technique with multi-criteria objectives. For instance, while a GDE attempts to minimize the time of completion of all tasks, another instantiation of the same GDE might have multiple objectives for minimizing the time of completion and the use of resources simultaneously. Because this entity assists the entire control architecture, it receives information from all the entities in order to gain a broader view of the system. For this reason, it runs the decision-making technique to optimize the operation in the control architecture. As a result, the GDE transmits the instructions to both LDEs and RDEs in order to improve global operations. In this case, these instructions involve a hierarchical relationship between the pairwise entities. Further information concerning adoption or non-adoption of the instruction and the type of relations between the pairwise entities is provided in the description of the reconfiguration mechanism. The decision-making technique for the GDEs are predictive techniques to guide the achievement of goals and support the optimality requirement of the control system. It is mainly, but not explosively, used in mathematical programming, metaheuristics, and big data analytics to improve the optimality in the system operations. Finally, for the performance indicator, the GDEs generally have metrics related to the completion and effectiveness exhibited in the system execution.

In general, the communication between the decisional entities, which according to a request-grant service or information protocol, helps to set the cooperation rules within the system environment. Figure 15 illustrates the communication between these entities. The interaction in the proposed approach are focused in the type of relation between entities (e.g. hierarchical, heterarchical, modified-hierarchical). However, even though the general protocol is illustrated, the parametrization of this protocol according to the defined interactions is out of scope of this dissertation. Then, the efficient interactions between the decisional entities are fully assumed in this dissertation.

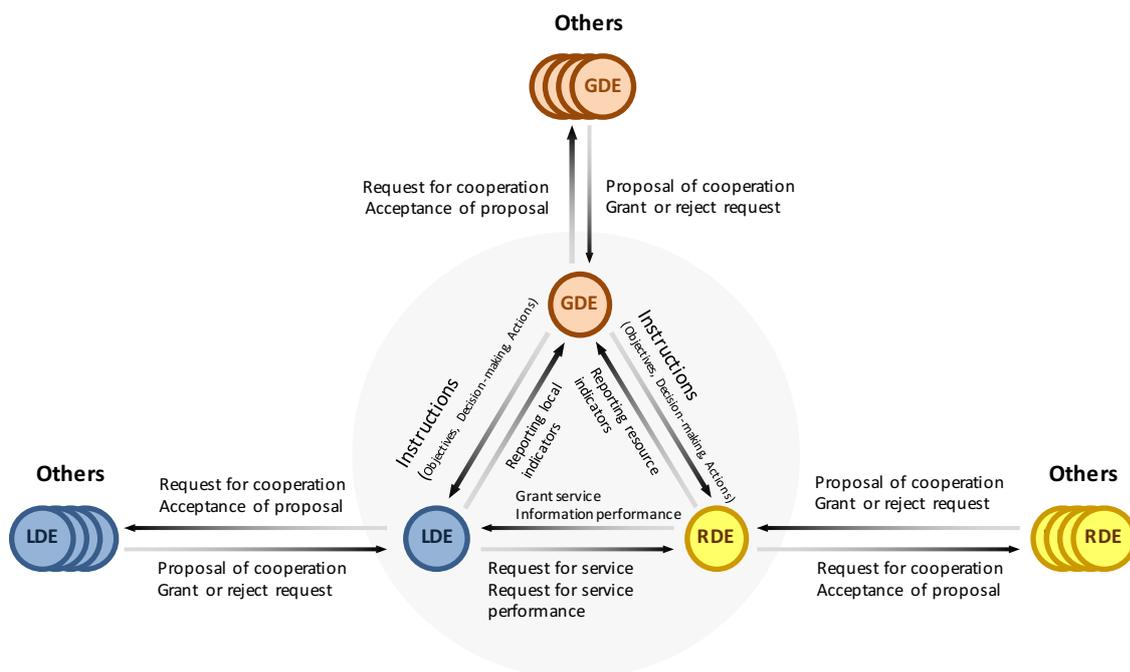


Figure 15 Communication protocol between decisional entities.

### 3.3.2.3. Layers of the control architecture

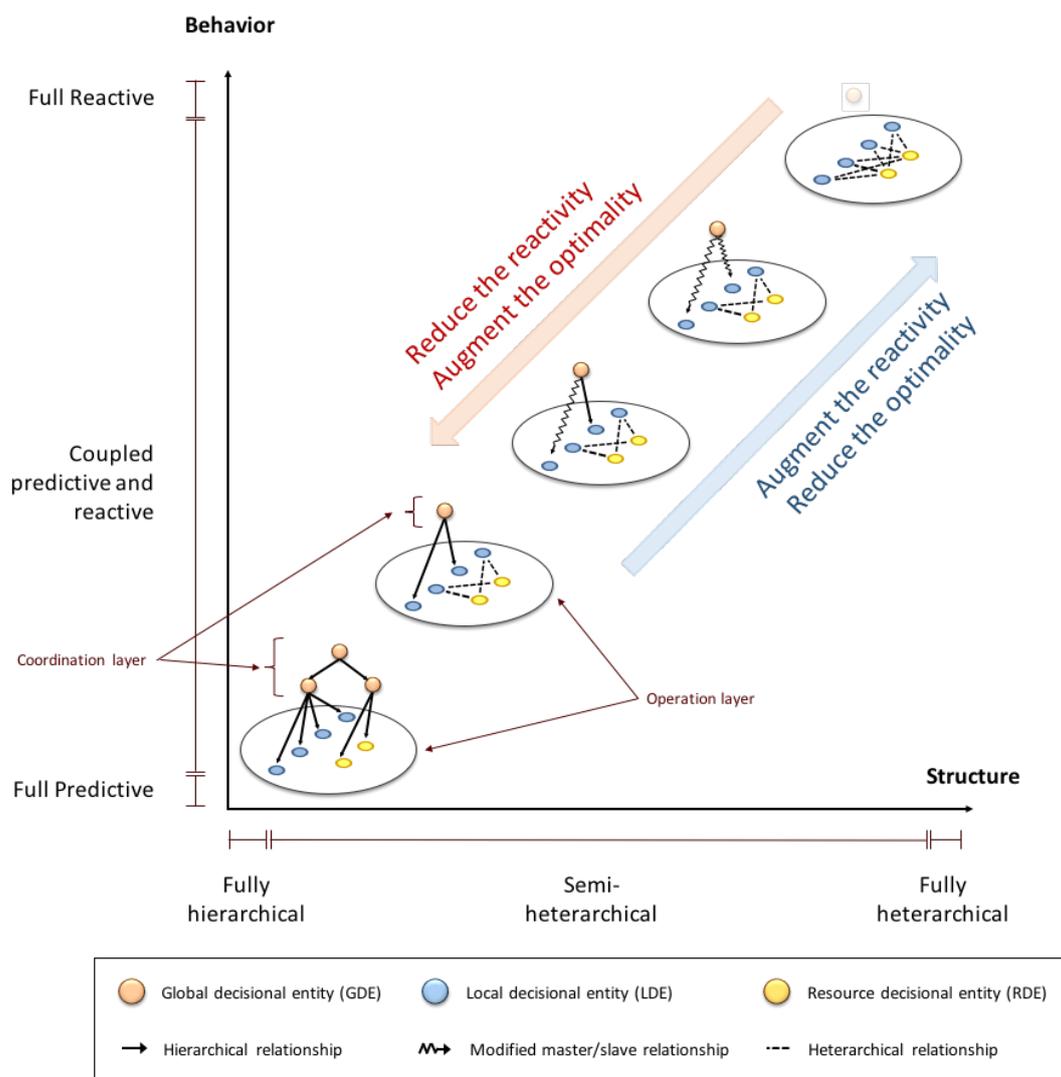
Pollux allocates the decisional entities either in the coordination layer or the operation layer (**Spec 3**). The coordination layer hosts one GDE or a set of GDEs that, either with hierarchical or heterarchical sub-structures and maintains hierarchical or modified master/slave relationships to assist the control system in meeting the global objectives. The operation layer hosts the LDEs and RDEs that, in heterarchical structures and eventually guided by GDEs, assist in the completion of the divided tasks of the global control problem. This consideration of two layers also responds to the technical implementation of this approach; while the coordination layer may be programmed in a centralized computer, the entities of the operation layer may be programmed in different computers in a heterarchical manner.

### 3.3.2.4 Structure and behaviour of the control architecture

Pollux features a flexible control architecture in terms of its structural and behavioural configuration. For Pollux, flexibility is defined as the capacity of a system to change and assume different positions or states to adapt to uncertainties (ElMaraghy 2006). The control architecture featured in Pollux provides a flexible composition that is capable of changing its organization (structural) and conduct (behavioural) during execution according to the system needs. This flexibility corresponds to the Pollux specifications (**Spec 4**). The structure can be tailored into a centralized, fully hierarchical, fully heterarchical or semi-heterarchical combination (see the X-axis in Figure 16). Furthermore, the decision-making techniques which influence its behaviour can be tailored based on predictive, reactive or coupled predictive-reactive techniques (see the Y-axis in Figure 16). This inner

flexibility aims to offer a larger set of configurations which, tailored by the reconfigurable mechanism, provides an improved control solution to the control system requirements.

Pollux uses the **governance parameters** of the decisional entity to define the structural and behavioural configuration of the control architecture (**Spec 4**). Although the governance parameters can define different attributes of the decisional entity in an extensive model of Pollux, the decisional entities of Pollux have at least a governance parameter that determines the relationship and interaction between two decisional entities. These specific governance parameters are the basis of the reconfigurability in Pollux, as a change in these parameters completely alters the control architecture of the system (i.e. both the structure and behaviour). The governance parameters define the type of relationship and the strength of this relationship. While the type of relationship refers to hierarchical or heterarchical relationship maintained between the two entities, the strength of this interaction refers to the degree of influence between the two entities.



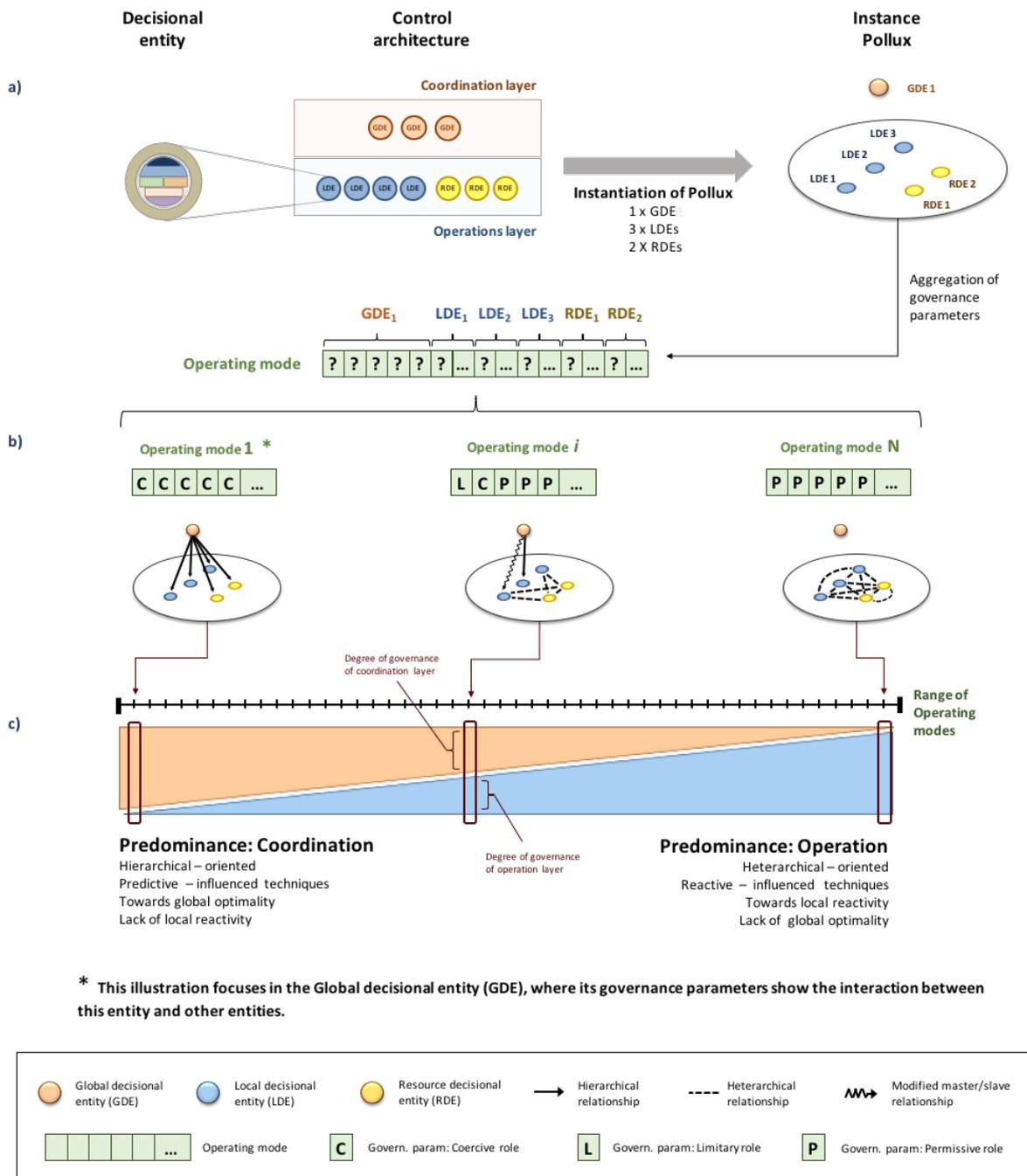
**Figure 16** Mapping of the possible configuration of the Pollux control architecture

A main characteristic of Pollux is that the interactions between two entities are not a binary attribute between hierarchical and heterarchical relations, but conversely, each has a degree of influence between these two relationships. This degree of influence ranges from full influence based on hierarchical relationships (e.g. master/slave) to no influence in heterarchical relations (e.g. cooperative), and may include intermediate and blended relationships in an interaction referred to as modified master/slave influence. This moderated influence refers to relationships where a master entity sends an instruction to a slave entity in terms of options or alternatives so that the slave entity has a partially autonomous decision between these bounded possibilities. An example of this type of interaction is the liminary interaction introduced in Zambrano Rey (2014a). Other types of modified master/slave relationships might be derivations of intermediate relationship, such as limitation in a maximum and/or minimum threshold, offering actions among a cardinal number of alternatives, or even selecting the objective or decision-making technique from a set of possible alternatives. Ultimately, the interactions of Pollux set the definitive behaviour of any entity. Full influence interactions will result with decisions obtained from predictive decision-making techniques, and null influence interactions will result with decisions obtained from reactive decision-making techniques, and modified master/slave interactions will be generally decisions reached by coupling predictive and reactive decision-making techniques.

### 3.3.2.5 Operating mode: representation of the control architecture

Considering the requirement of flexibility and customization of the control architecture, Pollux introduces a concept for representing, characterizing and customizing the control architecture. Inspired by previous approaches that have a pre-defined and reconfigurable system configuration (Monostori et al. 2015; Raileanu et al. 2012; Belisario & Pierreval 2013; Gumzej et al. 2009), Pollux introduces the *operating mode* as a key attribute of the control system that characterizes the objective and control strategy (**Spec 5**). An *operating mode* in Pollux is defined as a specific parameterisation (definition of all parameters) that establishes a specific setting of the control architecture (Jimenez, Bekrar, Trentesaux & Leita, 2016). The concept of operating modes, also called operational states by some researchers, is used to characterize different operational conditions of the system (Brdys et al. 2008). In addition, it reflects the states of the entire set of factors that influence the system's ability to achieve the prescribed control objectives (Monostori et al. 2015). Further information about the characterization of the operating mode is found in (Jimenez et al. 2015b)

The operating mode in Pollux is an organized array comprised of the governance parameters of the decisional entities. This representation array aims to have a unique representation that characterises the control architecture, evaluates the benefits of the control architecture in advance, distinguishes the control architecture capabilities, and provides some insight into the expected results. Therefore, the main advantages of the operating mode are that it provides a well-defined identification of the control configuration, enables evaluation of different alternatives for the reconfiguration process, and eases the change of configuration when necessary. The flexibility of the structure and behaviour of the control architecture, and the representation of a specific operating mode, are illustrated in Figure 17 below.



**Figure 17** Relation between the governance parameters and the operating mode. a) Example of control architecture, b) illustration of the representation of the operating mode through a structured array, and c) illustration of the range of operating modes

Figure 17a illustrates an application of a control system with a single GDE, three LDEs and two RDEs. This example focusses on illustrating the governance parameters of the GDE to illustrate the configuration of the control architecture, referred to as the operating mode. For this example, the GDE contains five governance parameters that define the interactions within itself and between it and each other decisional entity (i.e. first three for the LDEs and last two for the RDEs). For the values of these governance parameters, this example considers the coercive and limitary role of possible interactions between the decisional entities. While a *coercive role* corresponds to the direct instruction of actions (i.e. imposing an objective, behaviour or action) to be performed by the LDEs or RDEs, a

*limitary role* corresponds to a modified master/slave relationship in which the GDE proposes a set of alternatives for the LDE or RDE to permit these entities to make a decision within these boundaries. While coercive and limitary roles are based on the interaction concept proposed by Zambrano Rey (2014a), there is also a permissive role as a complementary interaction as distinct from the other two modes. In *permissive role* interaction, which encourages a heterarchical relationship, the GDE delegates full decisional autonomy to LDEs and RDEs according to its own objectives, decision-making and actions.

Figure 17b illustrates three different examples of operating modes. Depending on how the governance parameters are configured, the operating mode provides a simple identification of the structure and behaviour of the control architecture. This representation provides further information about the collective behaviour of the control architecture. In this sense, while some operating modes may comprise a fully hierarchically orientated architecture that collectively tends towards a predictive solution, others may be heterarchically orientated, i.e. a system that collectively tends towards a reactive solution. Nonetheless, Pollux encourages control strategies in which the operating mode contains a shared governance between the coordination and the operation layer.

Figure 17c illustrates the range of possible operating modes in a control architecture. This control constitutes shared governance between the coordination and operation layers. However, the status in the physical system and the objectives of the control system require steering the degree of governance between these two layers. In this sense, a reconfiguration mechanism must balance this governance of the control architecture.

### 3.3.2.6. Dynamics progression of the control architecture

The dynamics of the control architecture depend mainly on the operating mode. The execution process of the control architecture follows a general routine based on an interaction between coordination, request and bidding from the decisional entities. The procedures follow this general routine in order to resolve the global control problem. The means of execution depend on the operating mode. Figure 18 illustrates the dynamics of the control architecture.

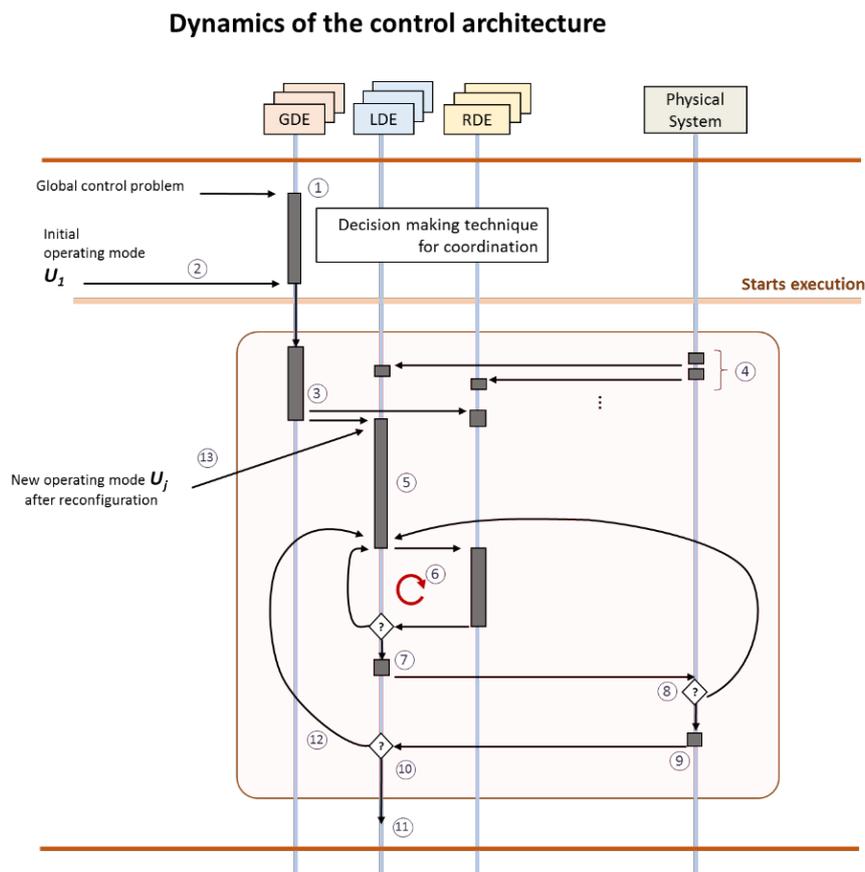


Figure 18 Dynamic process of the control architecture

### Dynamic process of the control architecture

#### Before execution starts (Offline)

1. GDEs execute and coordinate the actions offline through their own decision-making technique.
2. At the beginning of execution, The decisional entities (GDEs, LDEs and RDEs) are acknowledged with the initial operating mode (defining the governance parameters to start execution)

#### After execution starts (Online/Real-time)

3. GDEs instruction the actions to the decisional entities of the operation level (LDEs and RDEs).
4. The elements of the physical system report periodically the status to the decisional entities at the operation level.
5. LDEs executes the decision-making technique and starts acting according its own governance parameters.
6. LDEs start searching the completion of its tasks taking into account the service-oriented processing of RDEs. This loop ends when the LDE finds a RDE that will execute the next task.
7. LDEs actuates over the physical element and commands the physical actions to complete the task.
8. If the monitoring performed in step 4 finds that the element in the physical system starts the task processing, it waits until the task is completed. Otherwise, it start the search of RDE going back to step 6

9. The Physical element informs the completion of the task requested
10. LDEs evaluate if all the assigned task has been completed.
11. LDE has finish the task or sub problem to solve
12. If the LDE has still tasks to execute it returns to step 6 for its completion

The new operating mode  $U_j$  is implemented

After the reconfiguration, the process is the same but the outcome changes as the new governance parameters modifies the actions of the decisional entities.

### 3.3.3 Reconfiguration mechanism

#### 3.3.3.1 General description of the reconfiguration mechanism

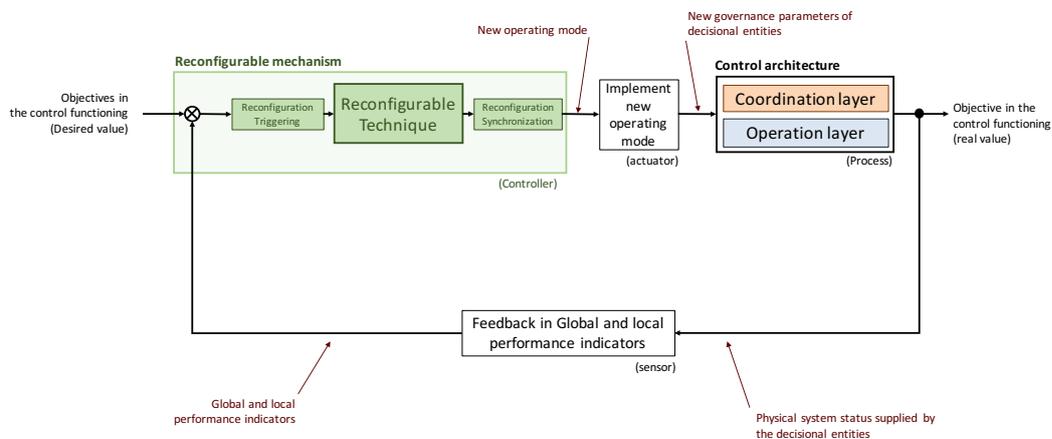


Figure 19 General process of the reconfiguration mechanism and its control over the control architecture

The reconfiguration mechanism in Pollux is dedicated to changing the configuration of the control system (i.e. the control architecture). To do so, it uses the operating mode (**Spec 5**) and the governance parameters (**Spec 4**) to change the control architecture. During the execution phase, this mechanism manages the reconfiguration processes and changes to a better suited operating mode in response to detection of an improvement or in response to a perturbation (**Spec 6**). As indicated above, an operating mode characterizes the control architecture and reflects the control solution and strategy for a given problem. The reconfiguration mechanism thus aims to continuously find an optimal operating mode and bring a change modifying the corresponding governance parameters. The general process of the reconfiguration mechanism and its control over the control architecture is illustrated in Figure 19. The general process of the reconfiguration mechanism is illustrated in Figure 20

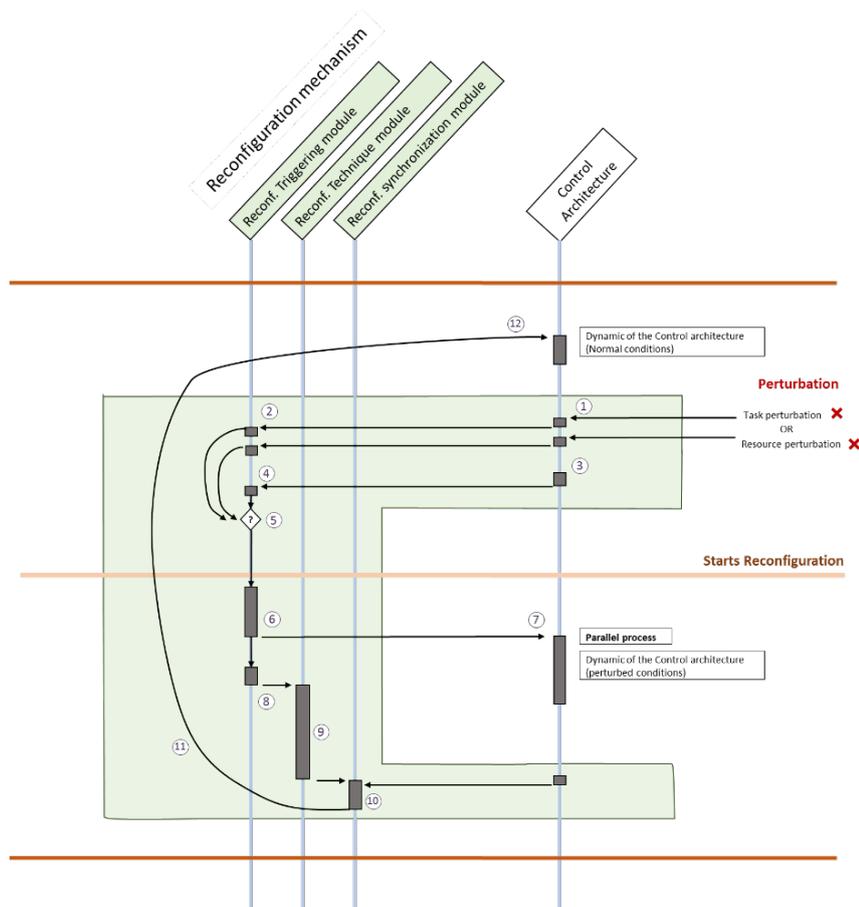


Figure 20 General process of the reconfiguration mechanism

### General process of the reconfiguration mechanism

#### *Before reconfiguration starts*

1. The physical system acknowledges a perturbation event from either the task or resource to the decisional entities of the operational layer (LDEs and RDEs).
2. The **reconfiguration triggering module** receives the desire of reconfiguration from LDEs and RDEs in a local monitoring process.
3. The decisional entities (GDEs) detect a perturbation from the global performance perspective.
4. The **reconfiguration triggering module** receives the desire of reconfiguration from GDEs in a global monitoring process.
5. The **reconfiguration triggering module** evaluates the intentions and trigger the reconfiguration when it is necessary.

#### *After reconfiguration starts*

6. The status of the control architecture is evaluated and it is set a default operating mode that continuous processing task according to its feasibility.
7. The control architecture starts its execution following the default operating mode
8. The **reconfiguration triggering module** requests the reconfiguration and supplies the requirements and setting of the reconfiguration process.

9. The **reconfiguration technique module** executes an improvement search algorithm to search a best suitable and optimal operating mode
10. Gathering the new operating mode and the current status of the control architecture, the **reconfiguration synchronization module** repair the action according to the deviation between the new operating mode expectancy with the real condition of the control architecture
11. The new and operating mode with the reparation actions are implement to the control architecture (Through the governance parameters)
12. The control architecture resumes the execution with the new operating mode.

### 3.3.3.2 Dynamics of the control architecture with the reconfiguration mechanism

The process of the reconfiguration mechanism changes the dynamics of the control architecture. This depends mainly on the interaction of the decisional entities in a fixed and static configuration. When the reconfiguration mechanism changes the operating mode and the governance parameters, this action changes the behaviour and execution in both the control system and the controlled system. To illustrate the dynamics of Pollux in terms of the control architecture and the reconfiguration process, Figure 21 shows the RCS from the request to resolve a global control problem to the final completion of the assigned task and activities.

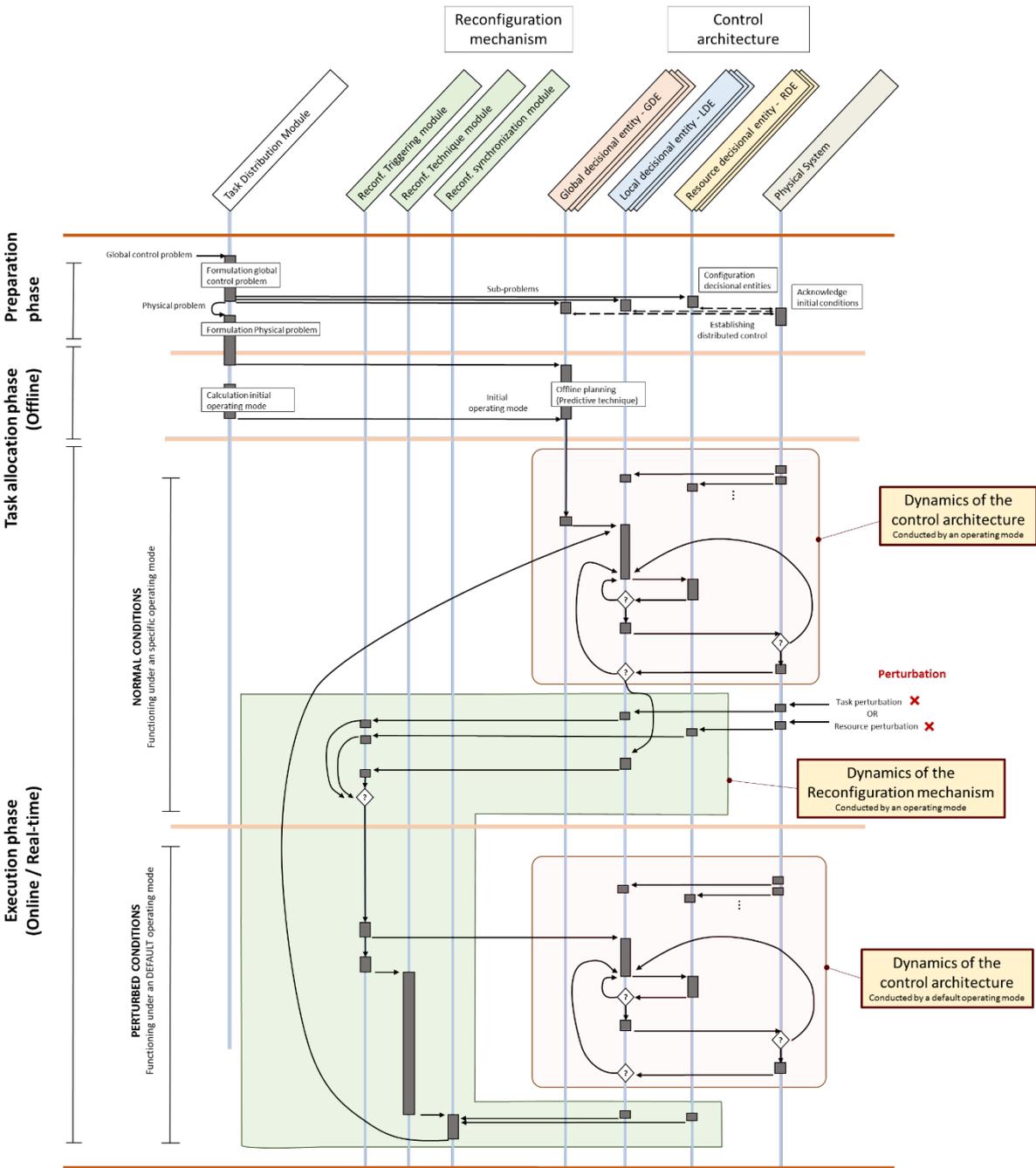


Figure 21 Sequence diagram of Pollux, including the preparation phase, task allocation phase and execution phase

Figure 21 illustrates the closed loop of the general process of the RCS. The process of Pollux is distributed in three phases: the preparation phase, the task allocation phase (offline) and the execution phase. In the preparation phase, Pollux divides the global control problem into sub-problems in order

to distribute the tasks among the available decisional entities. Hence, the decisional entities are configured, defining their components, including objective(s), decision-making techniques, parameters and governance parameters. At this stage, an initial operating mode is determined through the governance parameters according to the global control problem. Afterwards, in the task allocation phase, the GDEs plan the processing of the task in order to assign the specific actions to be executed by the LDEs and RDEs. Once this information is transmitted, the execution phase begins. During this phase, the reconfiguration mechanism monitors the execution control architecture in order to detect a deviation from the planned tasks. In the event of a perturbation, the reconfiguration process is initiated, the system enters a degraded condition and the control architecture is configured with a default operating mode. In parallel, the reconfiguration mechanism runs the reconfiguration technique in order to search for an optimal operating mode. At the end of this process, a new operating mode is provided, the system returns to its normal condition, and the control architecture resumes execution with the new operating mode.

The reconfiguration mechanism consists of three modules. These modules, functioning in series, are designed to respond to the stated challenges of the RCS. These modules, the reconfiguration triggering module, the reconfiguration technique module and the reconfiguration synchronization module, are described in more detail below.

**Reconfiguration triggering module:** this module, which answers the question “When to reconfigure?”, is responsible for monitoring the execution of the control architecture and triggering the reconfiguration process (**Spec 8**). The reconfiguration can be triggered by an adverse event such as deterioration in performance, or by a detection of a beneficial situation such as an improvement in system performance. In general, following previously established thresholds for global and local performance indicators (defined in the preparation phase), this module monitors the execution and the actual metrics to detect either a violation of thresholds or observed undesirable tendencies in these indicators. This module may 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 control architecture (reporting from decisional entities and the internal monitoring mechanism) and potential feedback from previous execution of the same component (i.e. learning capabilities). The outputs are the activation of the reconfiguration process, sending a reconfiguration triggering signal to the next reconfiguration module, and the information resulting from the analytics. Appendix E presents the pseudo-code of the reconfiguration triggering module.

**Reconfiguration technique module:** this module is responsible for performing the change of operating mode during the execution phase and finding a better operating mode (**Spec 7**). It is created to respond simultaneously to the questions “How and What to reconfigure?”. Based on the concept of an optimization process (Blum & Roli 2003), this module guides the process of searching for an optimal operating mode considering both the state of the system and a set of alternative solutions. In general, following a formulation of an optimization problem from the system state, the corresponding objectives and the reconfiguration parameters (e.g. maximum reconfiguration time, type of technique, etc.), it executes a method (generally iterative) that guides the search process to find a new configuration. This method aims to explore and select an alternative operating mode that responds to the reconfiguration objective (i.e. adverse or beneficial). Some examples of this technique include bio-inspired mechanisms, metaheuristics, combinatorial optimisation, neural networks and reinforcement learning. The inputs of this component are the current status of the control architecture (i.e. the control

system and physical system), the reconfiguration point provided by the triggering module, the results of the analytics and possible feedback from previous execution of the same module (learning capabilities). The outputs are the information from the guided search and the new operating mode to be implemented. Appendix E presents the pseudo-code of the reconfiguration technique module.

**Reconfiguration synchronization module:** this module is responsible for implementing the new operating mode in the control architecture (**Spec 9**). It helps answer the question “How to reconfigure?”; depending on the reconfiguration technique, the state at which it evaluates the implementation and the actual state when it is implemented may differ. In general, this module manages the deployment of the reconfiguration, as it adjusts the operating mode to ensure feasible implementation. This deployment is a post-optimization process that repairs the operating mode for a straightforward application. In this regard, it is necessary to monitor the execution of the system during default execution to obtain the real current state. The inputs of this component are the new operating mode, the information from analytics, the state of the system at the beginning of evaluation and the current state of the system. The outputs are the instructions to be followed to implement the new operating mode, which generally consist of the necessary changes in the corresponding governance parameters. Appendix E presents the pseudo-code of the reconfiguration synchronization module.

### 3.4 Reconfiguration technique for improving optimality and controllability

As previously stated, Pollux uses the reconfiguration technique module to improve the reconfiguration process. To this end, it is a framework with an iterative improvement search and uses optimization-based principles to improve the controllability and optimality of the reconfiguration (**Spec 6, Spec 7 and Spec 10**). Figure 22 illustrates the reconfiguration process following optimization-based principles.

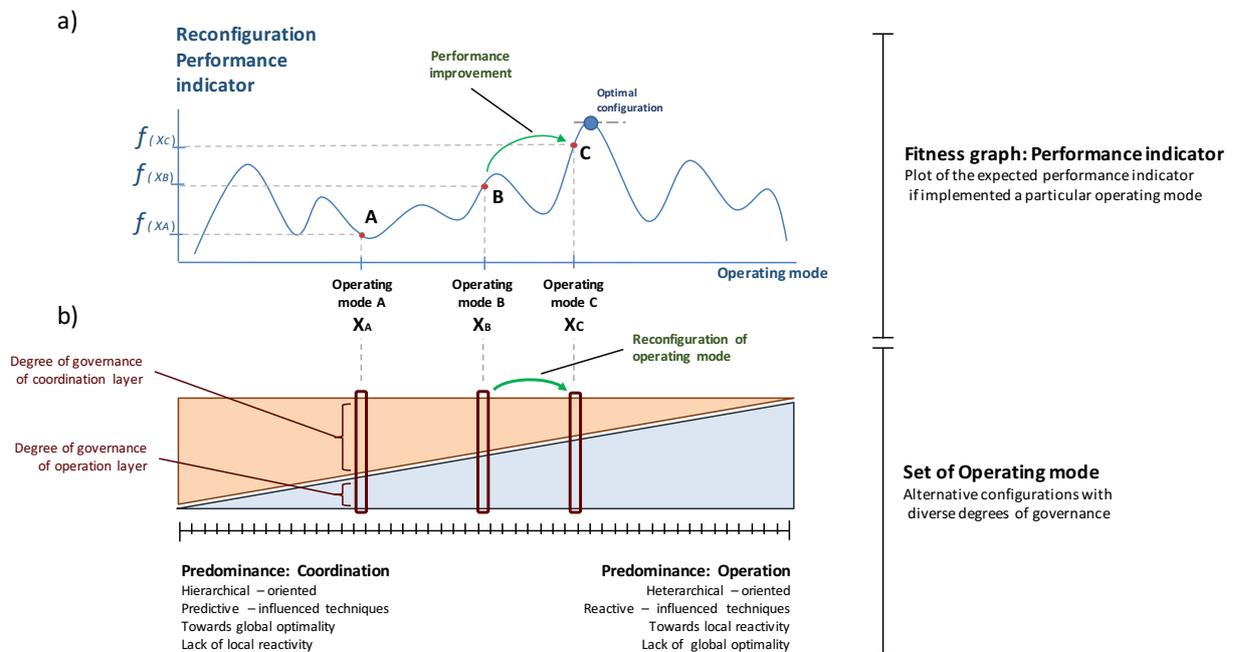


Figure 22 Process of the reconfiguration control technique to obtain the optimal configuration

The reconfiguration technique module starts when the reconfiguration triggering module detects a perturbation. From this moment, the reconfiguration process becomes an optimization problem and the control architecture enters into a default operating mode. In the reconfiguration process, each possible operating mode  $X_i$  characterizes a specific strategy and becomes a feasible alternative for resolving the problem. Figure 22 shows three different possible operating modes (plotted on the X-axis) with different degrees in the coordination-operation coupling (i.e. operating mode A or  $X_A$ , operating mode B or  $X_B$ , and operating mode C or  $X_C$ ). The reconfiguration process then evaluates the operating mode with the  $f(x_i)$  function (a function for the reconfiguration performance indicator) in order to diagnose the expected fitness result if such is applied. After this, the reconfiguration process runs an operation that searches for a better operating mode. The main advantage of this technique is its iterative improvement search process that strategically searches within different operating modes to find an optimal configuration. Depending on the degree of optimization required, based on the framework developed by Baker (1998), this technique may be optimal, near-optimal, towards-optimal or heuristic for searching within the selected subset. Finally, after these iterations, a better operating mode is provided and applied. Ultimately, following a repair process in the reconfiguration synchronization module, the new operating mode is implemented by changing the corresponding governance parameter within the decisional entities.

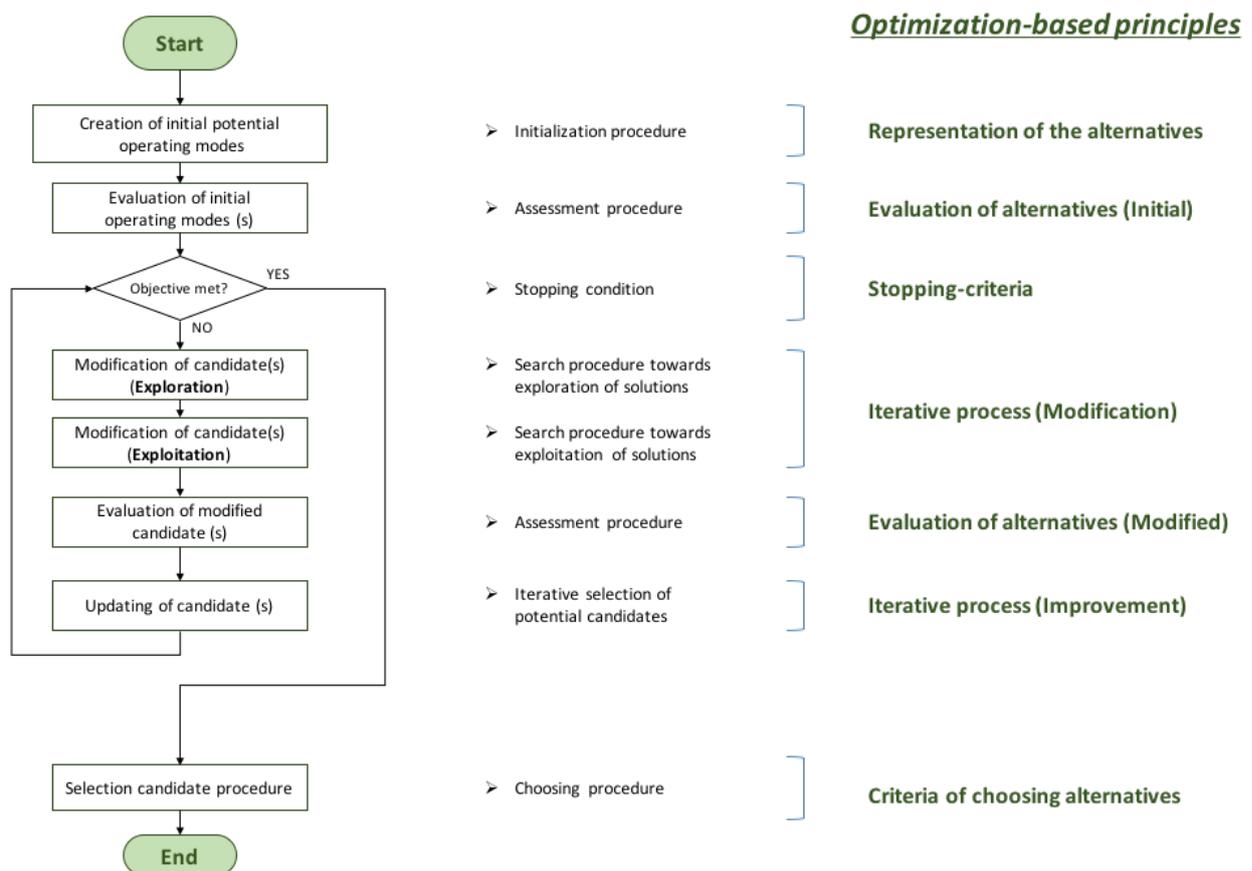


Figure 23 General process of the reconfiguration technique

Figure 23 illustrates the process in the reconfiguration technique module following the optimization-based principles (Spec 10). These optimization-based principles are included in the module as follows:

- **Representation of the alternatives:** The operating mode, which constitutes the aggregate of the governance parameters and reflects the configuration of the control architecture, is the representation of the alternatives that can be used to reconfigure the control system. In the reconfiguration technique process, the set of operating modes  $X_i$  includes the possible alternatives and represents the feasible solutions for the process in the reconfiguration technique module.
- **Evaluation of alternatives:**  $f(x_i)$ , the function of the reconfiguration performance indicator, is the assessment of alternative  $X_i$  in order to evaluate the expected performance if implemented. Although the function itself depends on the type of technique used in the reconfiguration process and the actual performance, this function is used as an estimation of performance during the reconfiguration process. Some examples of estimations include simulation indicators, algebraic functions related to the physical process, equivalent fitness functions to the physical process, or any other technique that evaluates performance.
- **Iterative process:** the module contains an iterative process that searches for a better operating mode in a repetitive evaluation of the alternatives. It starts with an operating mode as a current best alternative. Each time a new alternative (trajectory-based) or alternatives (population-based) is/are evaluated, the best alternative is changed if a new candidate is better in terms of performance. This process is repeated until the stop-criterion is met.
- **Stop-criterion:** The stop-criterion depends directly on the parameter set for the reconfiguration process. In general terms, it consists of the reconfiguration time. However, it may also be a satisfactory performance level, a degradation gap between the previous expected results, or an event in the physical system during the reconfiguration process.
- **Criteria for choosing alternatives:** After the iterative process, the module must choose a solution (a new operating mode to implement). In trajectory-based techniques this is straightforward because there is only one alternative. In population-based techniques, however, the result is a subset of feasible alternatives that can be applied. Generally, the best solution is chosen from the set, but other aspects, such as multiple objectives or feasibility issues, may be used as these criteria. Some examples in addition to simply choosing the best alternative include a second evaluation for a different objective, an average alternative for a moderate execution, and an alternative that fulfils a certain topological requirement, among many others.

### 3.5 Summary

This chapter described the external and internal view of the Pollux architecture. Table 4 summarize the characteristics of the proposed approach in terms of the static-features, in terms of the structural arrangement and behaviour functioning; and the dynamic features, in terms of the dynamic emergence and optimization capacity in the reconfiguration. This table positions our approach according to the literature of reconfigurable control systems reviewed in Chapter 2 (See Table 2).

**Table 4** Characterization of Pollux regarding the static and dynamic features of the architecture.

	Composition properties				Reconfiguration properties										General description	
	a	b	c	d	e	f	g	h	i					j	k	
a. Type of control																
b. Structure arrangement																
c. Behaviour conduct																
d. Reconfiguration changeability																
e. Purpose of reconfiguration																
f. Change type of reconfiguration																
g. Triggering location																
h. Quality of the reconfiguration																
i. Level of optimality																
j. Domain																
k. Technology																
<b>Pollux Architecture</b>	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	

In this chapter is created a reconfigurable control system that manages and adjusts optimally and in real time the functioning and architecture of a controlled system. Pollux takes inspiration from a proposed framework for the optimal reconfiguration of semi-heterarchical control systems. It stands over the idea that the optimal reconfiguration that minimizes performance degradation is encouraged on the change of configuration throughout the structural and behavioural characteristics. Therefore, Pollux is constructed over the orchestration of the control architecture guided explicitly over the change of the governance parameters.

To this end, the controllability and the optimization is intended through the models that compose the control architecture and the reconfiguration mechanism of this approach. On one side, the controllability is intended through the composition of the flexible and customizable decisional entities that permits to influence the emergence of the functioning to reach a desire performance level. On the other hand, it is included optimality-principles to the reconfiguration to reach the best configuration and minimize the degradation caused by perturbations. The next chapter implements the proposed approach to a general problem in the manufacturing domain, describe is the applicability in this domain, and illustrates in an example how is reached the controllability and optimality required characteristics.

# Chapter 4

## Application to the control of flexible manufacturing systems

### 4.1 Introduction

This chapter describes an application of Pollux for a flexible manufacturing system (FMS). The purpose is to present the applicability of the proposed approach in a specific domain. The implementation of Pollux within a FMS is chosen as a case study because it contains the flexibility needed for reconfigurability purposes and has the decision-making process before and during execution. In addition, the online configuration of this case study is seen as an optimisation problem. In general, the reconfiguration mechanism of Pollux can orchestrate the functioning of the control system as it can control and manage the configuration of the manufacturing control system online in order to improve the efficiency, effectiveness and reactivity of the FMS.

An overview of FMS and the specification of the flexible job shop problem (FJSP) are presented in section 2. Section 3 introduces the control objectives, system objectives, decision variables, parameters, assumptions and constraints of the FJSP as they relate to the case study. Finally, section 4 describes the Pollux implementation, detailing the parameterization of the external and internal view of the reconfigurable control system, including its components, control architecture and reconfigurable mechanism.

### 4.2 Overview of flexible manufacturing systems

A FMS is an assembly of computer-controlled machines and material handling devices that can simultaneously process medium-sized volumes of jobs with predefined characteristics (Browne et al. 1984). These systems are able to produce a variety of similar jobs as the production capabilities and layout can adopt different positions in response to changing requirements (De Toni & Tonchia 1998). The main advantages of implementing FMS are that it offers an improvement in the competitive value

of the manufacturer, an augmentation of productivity in a cost-efficient system, a decrease of setup time, a reduction of inventory, and an improvement in resource usage. However, the main drawbacks are that it requires a high initial investment for production operations and it may require an intelligent control system for implementation.

### 4.3 Description of the FMS control problem

The FMS control problem chosen for apply the proposed reconfigurable control system is the flexible job shop problem (FJSP). A FJSP, which is a generalization of the job shop and the parallel machine environments in scheduling problems (Pinedo 2012), provides a closer approximation of a wide range of real manufacturing systems. FJSP is a shop-floor system where jobs must follow a pre-determined operation-sequence and operations, and each operation can be processed by one or more machines. This system has two main objectives. It searches to assign each operation to a machine and searches to sequence the operations on the machines in order to optimise a certain objective or objectives (Marzouki & Belkahla Driss 2016). The following formulation describes the main specification of the FJSP for use in the FMS application. To this end, the FMS problem, which is the physical problem in this case study, and the control approach, which it is the implementation of the Pollux approach, are formulated. These formulations are detailed bellow.

#### 4.3.1 Formulation of the flexible job shop problem

The problem for this case study is the FJSP introduced above. The FJSP is an arrangement of machines with flexible capabilities that are able to produce a certain type of products with similar characteristics. The problem in this case study concerns the production of  $N$  jobs in the available set of  $M$  machines of the flexible manufacturing system. The description of the model is presented below.

The FJSP model is a graph  $G = (E, V)$  that forms the topology of the flexible manufacturing cell. The vertices are either machines or a turn-point (i.e. divergent or convergent vertices) within the cell. The edges are connections that transfer the jobs between the vertices. This adjacency matrix  $|V| \times |V|$  represent the partial connection and directed characteristics of the graph  $G$ . The manufacturing cell is composed of two types of resources: the machine  $M$  and transportation devices  $H$ . The cell consists of  $M$  Machines (or workstations) that are placed around the cell in a subset of vertices  $V$ . The transportation devices  $H$  are material handling devices that transport the jobs within the manufacturing cells guided by the edges  $E$ . In this problem, a job  $J$  is an order to produce a certain product. The cell is able to produce different types of products  $P$ . Each product has different configurations from the set of components  $C$ , and it has to be processed through an operation-sequence from the set of operations  $O$ . Each machine  $M$  processes a subset of operation  $P$  so that the entire set of machines  $M$  completes the manufacturing jobs. The set, parameters and constraints in this problem are as follows:

#### Sets and parameters

- $G$  : topology representation of the cell (duple)
- $V$  : set of vertices of  $G \{1, \dots, v, \dots, V\}$ ,
- $E$  : set of edges of  $G \{1, \dots, e, \dots, E\}$ ,
- $M$  : set of machines  $\{1, \dots, m, \dots, M\}$  and  $M \subset V$ ,
- $TP$  : set of turn-point  $\{1, \dots, tp, \dots, TP\}$  and  $TP \subset V$ ,

## Chapter 4. Pollux for a flexible manufacturing system

- $H$  : set of machines handling devices  $\{1, \dots, h, \dots, H\}$ ,  
 $OPE$  : set of operations performed by the cell  $\{1, \dots, o, \dots, OP\}$ ,  
 $P$  : set of product that can be produced in the manufacturing cell  $\{1, \dots, p, \dots, P\}$ ,  
 $O_p$  : set of list of sequenced operations that complete product  $p$   $\{1, \dots, o_p, \dots, O_{p'}\}$ ,  
  
 $R_{OPE}$ : set of machines  $m$  that process operation  $o_{PE}$   
 $MX_{vv'}$  : parameter matrix binary parameter with connection between vertices and  $v, v' \in V$ ,  
 $p_{om}$  : parameter of processing time of operation  $o$  in machine  $m$ , and  $o \in OP$ ,  
 $t_{vv'}$  : parameter transfer time between vertices (material handling devices  $H$ ) and  $v, v' \in V$ ,  
 $w$  : parameter maximum number of jobs simultaneously in the shop floor  
 $cb_m$  : parameter of the capacity of input buffer for machine  $m$   
 $J$  : set of jobs order to be produced  $\{1, \dots, j, \dots, J\}$ , and  $j \in P$ ,

This case study has the following assumptions:

### Types of assumptions

Related to the cell	<ul style="list-style-type: none"> <li>• The manufacturing system in cell with a continuous circulated flow of jobs</li> <li>• The cell permits that the jobs circulate through all vertices without blocking the flow of another job</li> <li>• All the machines are available to start production at time <math>t = 0</math>.</li> <li>• There is are <math>t</math> setting-up times for the machines</li> <li>• The loading and unloading machines may or not be processed in the same machine.</li> <li>• A machine (or workstations) consist of a single machine.</li> <li>• Each job is processed only by one machine at a given time</li> <li>• Machines can process the operation of only one job at the same time.</li> <li>• The machine is again available when the operation is completed</li> <li>• There are not pre-emption processes in the manufacturing cell.</li> <li>• The processing times of operations are deterministic times.</li> <li>• It is allowed recirculation of the product.</li> </ul>
Related to the job	<ul style="list-style-type: none"> <li>• The set <math>J</math> is the production order providing the products <math>P</math> to manufacture.</li> <li>• A production order is a batch of several jobs which the type and quantity of products to process.</li> <li>• Each job has a unique sequence, but this could be processed several routings from the combination of machines (flexibility in the production sequence)</li> <li>• All the job could start production at time <math>t = 0</math> (release time = 0)</li> <li>• A job, that is initiated and released to the manufacturing cell, starts in a specific machine (Loading machine and process)</li> <li>• A job, that is completed and need to be discharged from the manufacturing cell, goes to a specific machine (Unloading machine and process)</li> <li>• The job that gets to an occupied machine, waits in the input buffer position until the previous job leave the machine.</li> <li>• A job can only be processed by one machine at the same time.</li> <li>• The job need to follow a strict operation-sequence according to a list of operations.</li> </ul>

Figure 24 illustrates an example of Pollux applied to the flexible job shop problem. The topology of this example is a graph of 11 vertices and 12 edges, and it is defined in the matrix  $V \times V$ . The system has five machines, making it able to process the operations defined in the set OPE. The problem in this example is to produce jobs  $J$  based on the system capabilities.

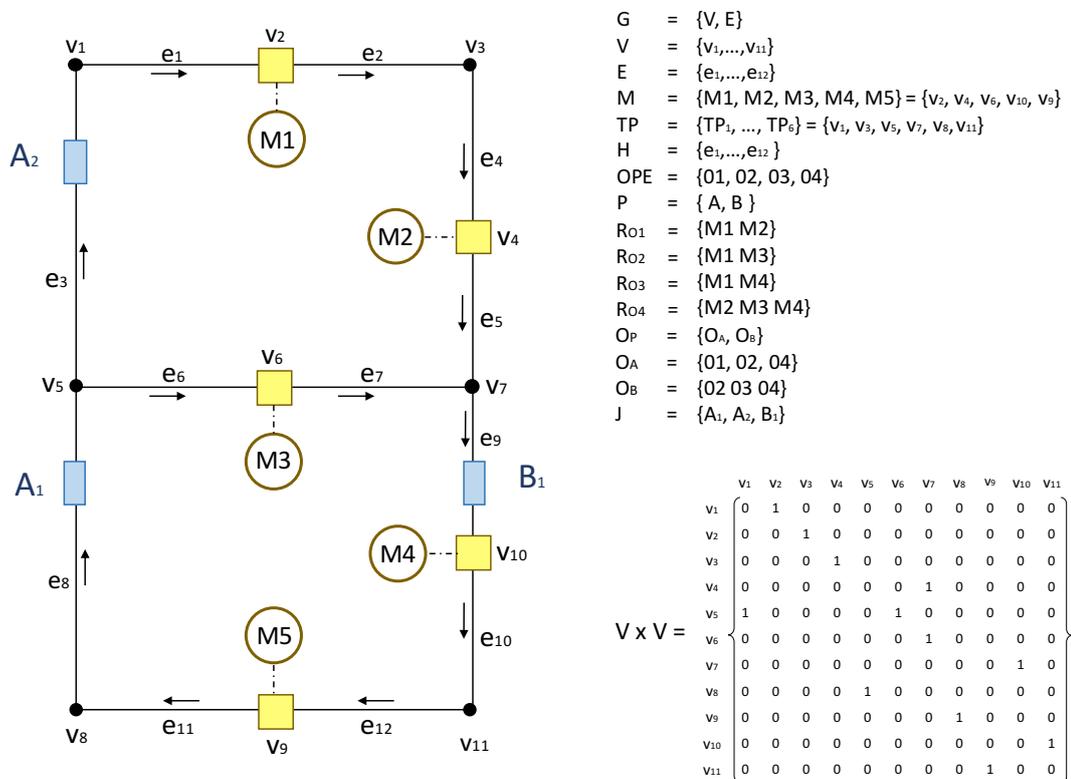


Figure 24 Example of a Flexible job shop system

### 4.3.2 Decision making in the Flexible job shop problem

In general, the decision-making in a flexible job shop problem depends on two factors: the complete execution of the production order, and the detailed instructions of the operations derived from the flexibility provided in the FMS.

For the first factor, the resolution of the problem requires deciding on the acceptance of the production order after an evaluation of the capability of execution and the planning in terms of use of the resources included in the cell. For this, planning and scheduling decision-making processes are required to provide the guidelines of production to completely execute the order during implementation. In short, the planning and scheduling allows the limited resources to be allocated to the activities in a way that it is optimised for the objectives and achieves the goals (Pinedo 2012). To this end, the production order is scheduled in the manufacturing cell based on the system parameters (i.e. processing times, transfer times, capacity of the cell, etc.) and then it is loaded for execution.

For the second factor, the resolution of the problem requires a decision on the alternatives provided by the flexibility of the manufacturing system, such as operation or routing flexibility (ElMaraghy 2006). In this case study, the FMS includes both machine-sequence flexibility and routing

flexibility. For *machine-sequence flexibility*, the jobs must follow a sequenced list of operations to complete the assigned tasks. Considering that the operations can be processed in a different machine, it must be decided, either in the planning, scheduling or execution phase, which machines will execute each of the operations. In this context, each job receives a machine sequence to follow during execution. With regard to *routing flexibility*, the job is constantly being transferred from one machine to another. Considering different edges that conform to different routes between machines and the recirculation attribute of the cell, the job has several paths to execute the transport between two machines. Thus, each job receives a route path to follow during execution.

These two factors influence the decision variables used in the flexible job shop problem. These decision variables are important in the execution because the decision-making technique must select, for each decision variable, an alternative from among the possibilities. In fact, this selection decision must be made regardless of whether the decision-making process is predictive or reactive. Thus, the division of activities from the global control problem must lead to a distribution of tasks according to the resolution of these decision variables. For the FJSP application, the decision variables are as follows. First, the **release sequence of jobs** to the manufacturing cell must be decided on. This type of decision variable refers to the order in which jobs get to the cell. Second, if a job is initially released or it has just finished an operation in a machine, the **subsequent machine to address** for that job must be decided on. Once the job is transported through a material handling device, each job must have a decision variable that defines the destination machine in which the next operation will be executed, as these can be processed by several machines. Third, in a type derived from the previous type of decision variables, a **machine sequence of jobs** must be defined. This type of decision variable takes into consideration that each job passes through a sequence of machines that, due to a combinatorial sequence, must be taken into account in order to optimise the processing execution. Fourth, considering the different routes in the cell, each job must decide on the **turn-point direction that the job follows**. This type of decision variable defines the way in which the job will be directed each time that it passes through a turn-point. Fifth, in a type derived from the previous type of decision variables, the **routing path of jobs** within the cell must be defined. Likewise, concerning the machine-sequence decision variable, this type of decision variable refers to the combinatorial routes that each job has when it is addressed to a new destination (i.e. successive turn-point). Finally, the sixth type of decision variable defines which operations are processed when the jobs arrive at a new machine. This definition is needed, as it might be possible to process a new operation in the same machine when the previous operation is completed, if the new operation has not been decided on.

Nonetheless, two issues must be considered for the decision variables. First, these types of decision variables may be attended and resolved by predictive or reactive decision-making techniques. However, due to the characteristics of the decision variable, in a distribution of sub-problems some of them could be assigned in a straightforward manner. For instance, while the routing path of jobs may be assigned to a predictive technique, the subsequent machine to address may be assigned to a reactive technique. Second, there is a connection between these types of decision variable. In fact, considering the order in which the decision variables are taken (i.e. first release, then machine and then routing), some decision variables are influenced by previous decisions. Therefore, later decision variables cannot actually decide on a set of alternatives as previous decisions limit their possibilities; in the worst-case scenario, it cannot decide anything and only reacts subsequently.

### 4.3.3 Performance indicators for the flexible job shop problem

The performance of a flexible job shop system may be measured according to the assigned problematic of a decision-making technique. Therefore, considering that the FJSP global problem distributes the tasks throughout the planning and scheduling of the production order, and the individual execution of each assigned activity, the associated performance indicators may include both absolute metrics at the end of the whole execution (i.e. global indicators) and relative metrics at particular time intervals during the execution (i.e. local indicators). Of course, these two approaches may be used for purposes other than being an effectiveness indicator, such as for completeness of tasks or as an efficiency indicator for resource consumption.

This example of Pollux evaluates only the effectiveness of the execution. This involves measurement of the effectiveness of the flexible job shop system in terms of the static and dynamic performance indicators. **Static Indicators:** some of the most frequently used indicators are the makespan, the total completion time and the total weighted completion. These indicators, for which equations are presented in table 4, aim to provide an absolute and static measurement that shows the total performance regardless of the variations that may occur during execution, such as **the makespan** indicator. Given that this measurement does not provide information about the events during execution, certain indicators may be used to provide a general idea of such information. The **total completion time** and **the total weighted completion** recognize that the tardiness of the jobs is not efficient for the execution. To this end, an aggregated evaluation is calculated that, based on the completion of each job separately, penalizes a weighted or non-aggregated function in proportion to the lateness of the job. **Dynamic indicators:** these indicators are characterized by providing a metric in relation to the time of execution. They aim to provide a measurement that gives information about the behaviour of the execution and in this way, monitor the execution. Two examples of this type of indicator are the throughput and the utilization per period (included in Table 4). While the **throughput** refers to the number of jobs completed in certain time window (e.g. 45 jobs per hour), the **utilization per period** refers to the consumption of a certain resource, such as machine usage time or material consumption (e.g. 20 minutes of machine processing per hour). Although these individual indicators are static during the period of the execution, a new time window changes the indicator and permits the evaluation of improvement or degradation. Certainly, an historical view of these indicators might contribute to monitoring the trajectory of system execution.

**Table 5** Static and dynamic indicators for the manufacturing domain.

Performance Indicator	Function	
<b>Static Indicator</b>		
<i>Makespan</i>	$C_{max} = \max_{\forall j} (c_j)$	Eq. 1
<i>Total completion time</i>	$TC = \sum_{\forall j} c_j$	Eq. 2
<i>Total weighted completion time</i>	$TWC = \sum_{\forall j} w_j c_j$	Eq. 3
<b>Dynamic Indicator</b>		
<i>Throughput</i>	$TP_i = \sum_{\forall j   t_i \leq c_j < t_{i+1}} u_j ; \forall i$	Eq. 4
<i>Utilization per period</i>	$UR_i = ut_{i+1} - ut_i ; \forall i$	Eq. 5
<i>Energy consumption</i>	$EC_i = \sum_{\forall m} e_m p_{mi} ; \forall i$	Eq. 6
<b>Conventions</b>		
$C_{max}$	Completion of the last job of the production order	
TC	Total completion time	
$c_j$	Completion time of job $j$	
TWC	Total weighted completion time	
$w_j$	Weight penalty attributed to the completion time of job $j$	
$i$	Time window or timeframe $i = \{1,2,3,4,\dots\}$	
$TP_i$	Throughput for the time window $i$	
$u_j$	Binary variable of completion of job $j$	
$t_i$	Starting point of time window or timeframe $i$	
$UR_i$	Utilization of resources during time window or timeframe $i$	
$ut_i$	Available quantity of resource at the beginning of time window or timeframe $i$	
$EC_i$	Energy consumed by cell during time window or timeframe $i$	
$e_m$	Energy consumed by machine $m$ per time/unit	
$p_{mi}$	Proportion of energy consumed by machine $m$ in the time window or timeframe $i$	

Table 5 indicates some of the performance indicators that allow evaluation of the execution of the cell for both the global problem and the corresponding focused sub-problems. On one side, the global indicators (Eq.1, Eq.2 and Eq.3) measure the effectiveness of the execution and enable measurement of the variability of completion during the execution time. On the other side, the focused indicators (Eq.4 and Eq.5) measure the efficiency of the jobs in an aggregated function within a time window. It is expected that the synergy between these indicators will make it possible to improve the controllability of the control system.

### 4.4 Pollux applied to flexible manufacturing systems

Based on the description presented above, this section customizes Pollux in an application to the FMS. First, this section presents the external (i.e. purpose and environment relation) and internal view (i.e. structure, behaviour and dynamism) of the control architecture. It then describes the reconfigurable control mechanism and the functioning in respect to the architecture. The description in this section is *generic*, as Pollux is created as a reference reconfigurable control system of semi-heterarchical architectures that may be tailored to the level of controllability and optimality needed in the control system. Nevertheless, this case study customizes the Pollux architecture in the FJSP and provides suitable models to use in the implementation. The architecture of Pollux applied to the flexible manufacturing system is illustrated in Figure 25

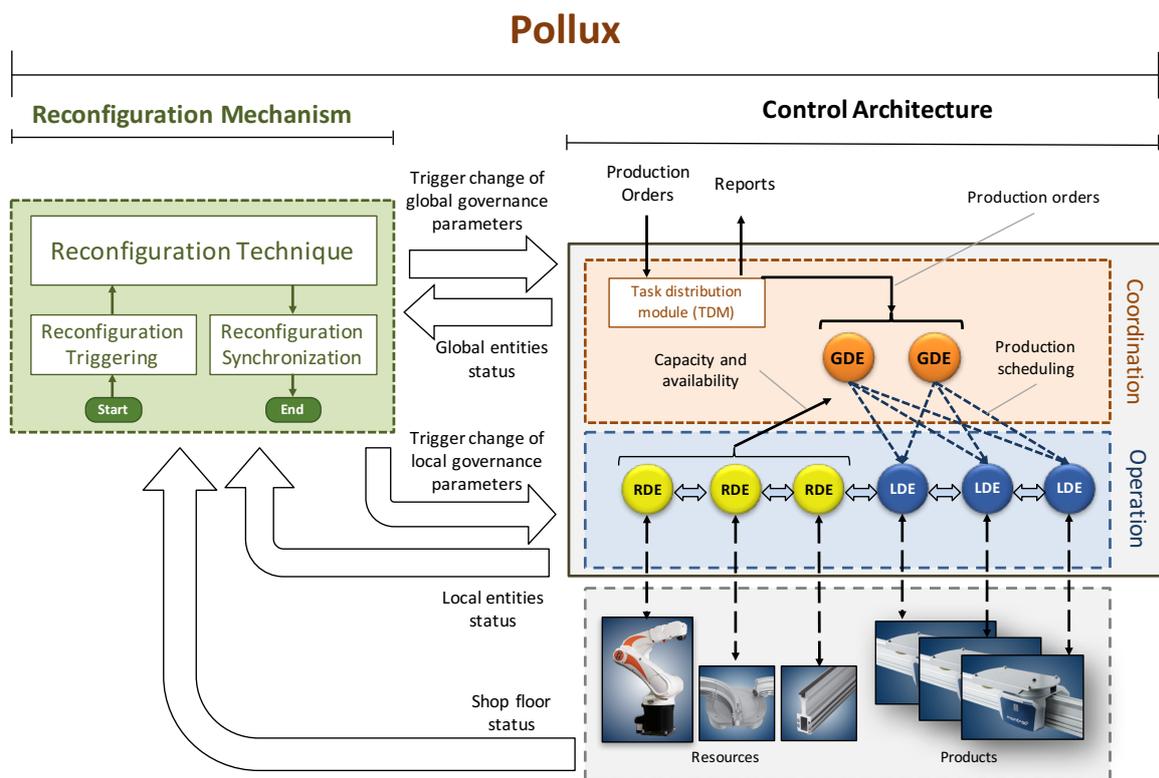


Figure 25 Architecture of Pollux applied to the manufacturing system.

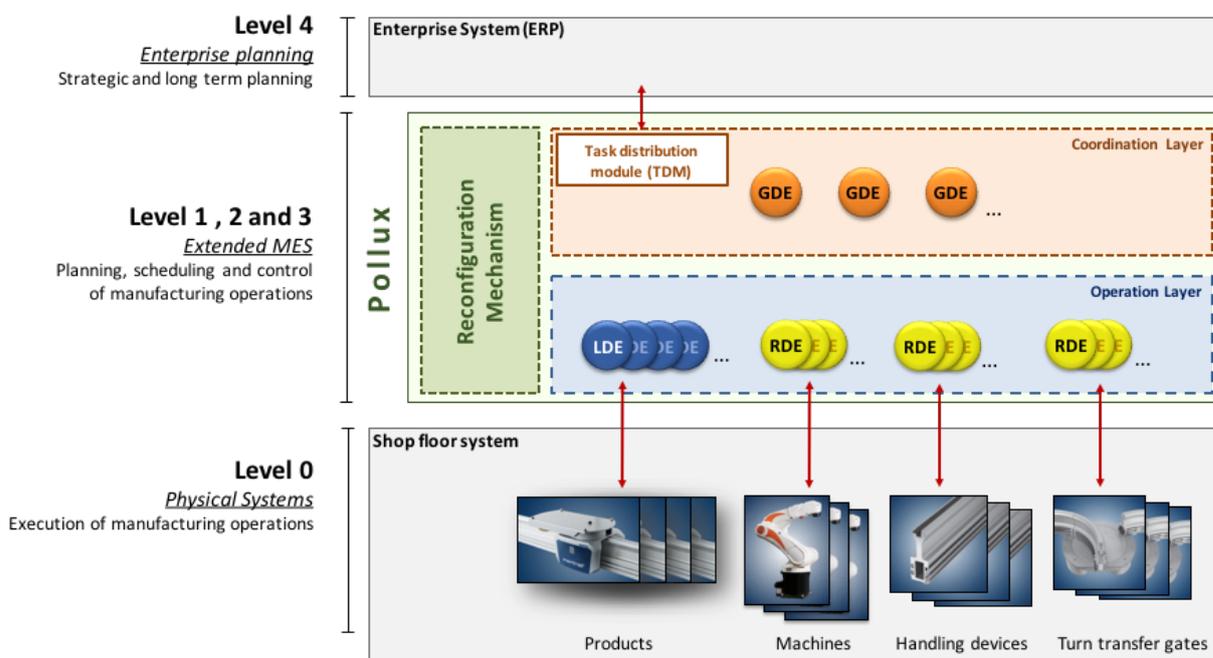
#### 4.4.1 External view

##### 4.4.1.1 Purpose

The main purpose for this example is to ensure the completion of the production order in an execution that minimises the makespan and responds to external perturbations. In the event of perturbations, Pollux must also minimise the degradation caused by the unexpected perturbation event through an optimal reconfiguration of the control architecture.

### 4.4.1.2 Environment interactions

In the case study framed under the manufacturing domain, Pollux is used as an extended manufacturing execution system (extended MES) that it is responsible for the planning, scheduling and control of the manufacturing operations in the flexible job shop system. The manufacturing standard ISA-95 for developing the automated interface and posterior functioning between an enterprise and control system divides the production planning, scheduling and control into five levels: physical system (level 0), sensor and actuators (level 1), SCADA (level 2), manufacturing execution system (level 3), and enterprise resource planning (ERP) (level 4) (Trentesaux & Prabhu 2013). Therefore, in order to position the scope of Pollux in the manufacturing system, the extended manufacturing execution system comprises the functioning and control of level 1, level 2 and level 3 as defined in the ISA-95 standard. The main advantage of locating Pollux between the ERP and the physical system (i.e. manufacturing cell) is that the extended manufacturing execution system takes over the decision-making processes on the shop floor, improving the global controllability and optimality of manufacturing operations.



**Figure 26** Location of the proposed approach within an industrial enterprise system (External view)

Figure 26 illustrates the location of Pollux within the manufacturing system of an industrial enterprise. The environmental relationship of Pollux is mainly with the ERP system and the physical system (i.e. manufacturing cell). On the one hand, Pollux obtains the alignments in manufacturing operations from the ERP, such as the instructions for the production order, quantities, types of products, expected production dates, quality specifications, expected receipt of materials or scheduled product transfer, and other information. At the same time, Pollux provides information to the ERP about the status, performance and capacity of the manufacturing cell in terms of the physical execution, resource capacity, quality performance, operational performance and maintenance schedule. On the other hand, Pollux provides distributed control of the constituent elements of the physical system (i.e. manufacturing cell). To do so, Pollux assigns a decisional entity from the operation level that, through sensors, obtains information and monitors the actions and behaviour of the physical element. At the same time, using actuators, the distributed decisional entities of Pollux conduct the activity of the

physical element according to the coordinated actions of the control system. In case of an unexpected event, both the ERP and the physical system detect and communicate the perturbation to Pollux in order to start the reconfiguration process. Then, while the ERP may recognize rush orders or expedite orders, the physical system acknowledges resource failures or major deviations from expected behaviours.

### 4.4.2 Internal view

#### 4.4.2.1 Structure arrangement and behaviour functioning

The internal view of this example is comprised of the control architecture and the reconfiguration mechanism. The control architecture, the semi-heterarchical architecture with flexibility capability, is responsible for virtually modelling the FJSP to conduct the planning, scheduling, execution and control of the manufacturing operations. This architecture is divided into the coordination layer, which contains the task distribution module for distributing the tasks of the global control problem and the GDEs for coordinating the behaviour in the manufacturing operations, and the operation layer, which contains the LDEs and RDEs that represent, respectively, the jobs and manufacturing resources in the distributed control of these operations. On the other hand, the reconfiguration mechanism is the process that changes the control architecture to respond to the unpredicted events in the manufacturing operations. This mechanism is positioned in a sequential arrangement of three modules in charge of the triggering, reconfiguration technique and the posterior synchronization of the reconfiguration process. Figure 27 illustrates the general arrangement of Pollux from an internal view.

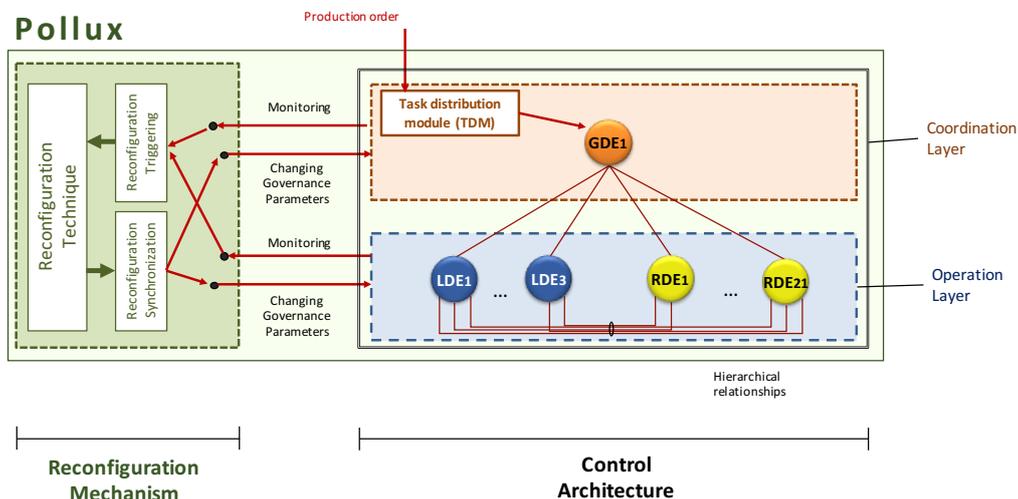


Figure 27 Structure of Pollux for the FJSP (Internal view)

#### 4.4.2.2 Coordination layer for the FJSP

##### Task distribution module

In this example, the task-distribution module (TDM) partitions the flexible job shop problem (i.e. the global control problem) and distributes the resulting sub-problems between the decisional entities. Initially, it gathers both the request from the ERP level (level 4) and the current capacity and status of the manufacturing cell (level 0) to formulate the control and physical problem of the FJSP example. It distributes afterwards the task between the decisional entities. In general, the global decisional entities (GDEs) are configured to resolve the planning and scheduling of the FJSP offline, as well as to monitoring the global performance indicators during execution. Each local decisional entity (LDE) is configured to control the specific job, is responsible for selecting the course of action of this job within the manufacturing execution, and evaluates its own performance. Each resource decisional entity is configured to control a machine, a material handling device or a turn-point device, is responsible for executing the processing (i.e. production, transfer or direction-guiding), and also evaluates its own performance. Readers can confirm the distribution of the tasks and the redundancy in the decision-making techniques. Indeed, although the GDE evaluates and selects the decision variables within the cell, the LDE and RDE also select the decision variables in the execution in a distributed manner. Nevertheless, the actual decision contained in the decision-variable is made according to the guidelines presented in the governance parameters.

The TDM also divides the problem according to the decisional variables. In the physical problem, the decisional entities resolve these decisional variables as this is included in the decision-making techniques. Nevertheless, in the control problem, the interactions between the decisional entities depend on the influence on these decisional variables. In this sense, the division of the problem in terms of decisional variables in the control is between the release order, machine-sequence and the routing path for each job. The release order specifies when each of the jobs is released, the machine sequence is an array of machines that will be guided during the job, and the routing path is the track followed in the material handling machine to get to the pre-defined machines.

### **Global decisional entities**

For the configuration of each GDE, the objective, the decision-making technique and governance parameters must be specified. The objective(s) of the GDEs may be related to the expected efficiency or effectiveness in the manufacturing execution. For example, some objectives aim to provide information that explains the performance of the execution, such as makespan or total completion, while others aim to provide the performance in relation to an expected execution, considering the due date of the production order, such as total earliness or total tardiness. For the decision-making techniques, the GDEs can be configured with predictive techniques that resolve the FJSP. Examples include genetic algorithms, exact methods, branch and bound methods, among many others. In fact, several approach for resolving the FJSP can be found in the operational research domain (Luke 2009). For the governance parameters, the decisional entities contain the method of interaction with each other decisional entity. The governance parameters from entities on the coordination level can be: coercive, as in master-slave relationships; limitary, as in modified master/slave relationships; and permissive, for null-influence relationships. Conversely, the governance parameters from the entities of the operation level may be: submissive, as when performing instructions commanded from upper entities; propositive, when offering actions or services to other entities; cooperative, for establishing collaboration, bidding or contract processes with other entities; and null, for entities that do not communicate.

In this example, a single GDE is configured at the coordination level, called GDE<sub>1</sub>. The objective of this decisional entity is aligned to the objective assigned by the ERP, as it seeks to minimise the makespan. The completion of the production order is implied in the minimisation of the makespan, and the minimisation of degradation is allocated to the reconfiguration mechanism. There is no need for more GDEs. For the decision-making technique, it might be configured with a genetic algorithm to solve the problem offline. The genetic algorithm creates a population of chromosomes as possible solutions to the FJSP that define the selection in the decision variables (i.e. release order, machine-sequence and routing path). As a result, this predictive technique plans and schedules the production order in the manufacturing cell, and it is ready to send these decision variables to the LDEs and RDEs for execution. The governance parameters of GDE<sub>1</sub> correspond to the interaction of the GDE with the LDE and the RDE for each job and resource in the cell (i.e. coercive, limitary and permissive interaction for 3 jobs and 21 resources). In this particular case, all governance parameters are set to coercive interaction to follow the predictive planning and scheduling. Figure 28 illustrates the global decisional entity.

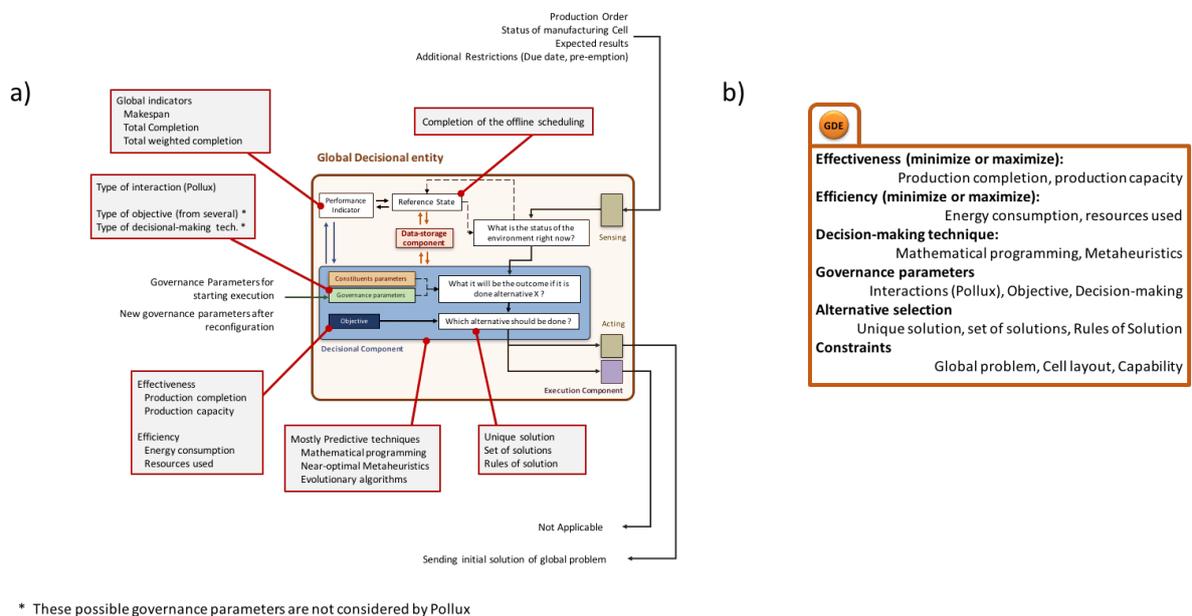


Figure 28 Illustration of global decisional entity GDE a) composition, and b) functional diagram

#### 4.4.2.3 Operation layer for the FJSP

##### Local decisional entities (LDE)

For the configuration of each LDEs, the objective, the decision-making technique for imposed and autonomous functioning and the governance parameters must be specified. The objective(s) of the LDEs may also be related to their own expected efficiency or effectiveness in the manufacturing execution. This must be tailored to the job performance during the execution. A good example of this type of objective can be found in Trentesaux and Giret (2015). These objectives may be related to expected partial or full completion, earliness or tardiness compared to expected completion, the consumption of resources such as energy or materials, or any other indicator of job performance. In these decisional entities, the characteristic of the goal-oriented entity is most frequently used in the decision-making technique.

The LDE has two different decision-making techniques that are used in parallel depending on the values in the governance parameters; these are default and imposed techniques. In fact, although the LDE configuration permits inclusion of other decision-making techniques, it must contain at least a technique for the default behaviour and the imposed behaviour. The default technique uses the goal-oriented protocol that searches the actions for its completion. Considering that the action is in response to the decision variables in the release order, subsequent machine to address and turn-point direction that the job follows, examples for this technique are first available machine, potential fields, negotiation protocols, closest machine, and others. The imposed technique is when a decisional entity from the coordination level sends a coercive or limitary instruction to the LDEs to follow during the execution. While the coercive instruction imposes a specific action or actions on the LDE to follow during the execution, the limitary instruction permits the default behaviour to conduct the action but the decisions are bounded by certain limitations.

Each LDE has a set of governance parameters relating the interaction with the rest of the decisional entities. For the interaction with entities of the coordination level, each LDE has a governance parameter for each GDE. This governance parameter, as the role that the LDE has regarding the interaction with the GDE, can be submissive for coercive interaction of the GDE, or autonomous for the limitary and permissive interaction of the GDE. For the interaction with other entities with the operation level, each LDE has a governance parameter for the interaction with other entities (i.e. LDE or RDE). This governance parameter, as the role of the LDE in regard to the same level relationship, can be propositive for offering the completion of an action if another entity accepts, or cooperative for sustaining a negotiation process with the other entity.

In this example, three LDEs are configured in the operation level, called LDE<sub>1</sub>, LDE<sub>2</sub>, and LDE<sub>3</sub>. The objective of this decisional entity is to complete the processing of operations of the assigned job. There is no need for more LDEs. For the decision-making techniques, while the default technique is configured with the protocol to evaluate the closest machine (i.e. selecting the next machine and routing path), the imposed technique is the protocol for accepting and executing the action imposed from the entities in the coordination level. Note that in this case there is no interaction between the LDEs, as they communicate only with the resources. The governance parameters of the LDEs correspond to the interaction with the GDE and the RDE. In this particular case, it is configured with a governance parameter with a submissive interaction for the GDE and 21 governance parameters with null-relationship for each of the resources. Figure 29 illustrates the local decisional entity

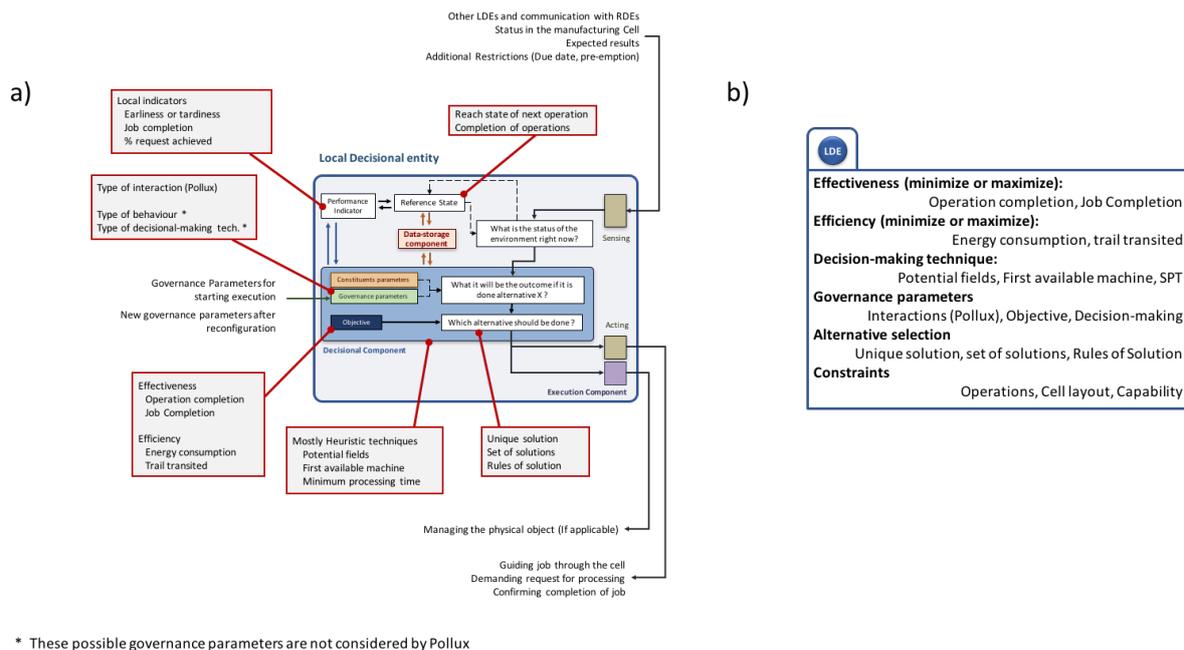


Figure 29 Illustration of local decisional entity LDE a) composition, and b) functional diagram

### Resource decisional entity (RDE)

The objectives, the decision-making techniques and governance parameters of RDEs depend on the type of resource to model from the physical system. This can represent assembly machines, material handling devices, auto-guided vehicles, turning gates, or any other device that can move or transform the manufacturing job. In general, RDEs are different from LDEs in that instead of looking to execute a specific assigned task, they provide a service whose purpose is generally to fulfil the other decisional entity’s task. Nevertheless, in addition to this service-oriented functioning, it must execute the requested services while it fulfils its own objectives. In this sense, the objective(s) of the RDEs are definitively related to the expected efficiency or effectiveness of its own service. For example, the RDE may have an effectiveness objective of fulfilling the requested services from the LDE (i.e. processing the requested operation). In terms of efficiency, the RDE might seek minimisation of the energy used, the raw material used for the manufacturing, or setup times, among other things. Either for efficiency or effectiveness, the objective is to provide the service needed from the other decisional entities (i.e. GDE and LDE).

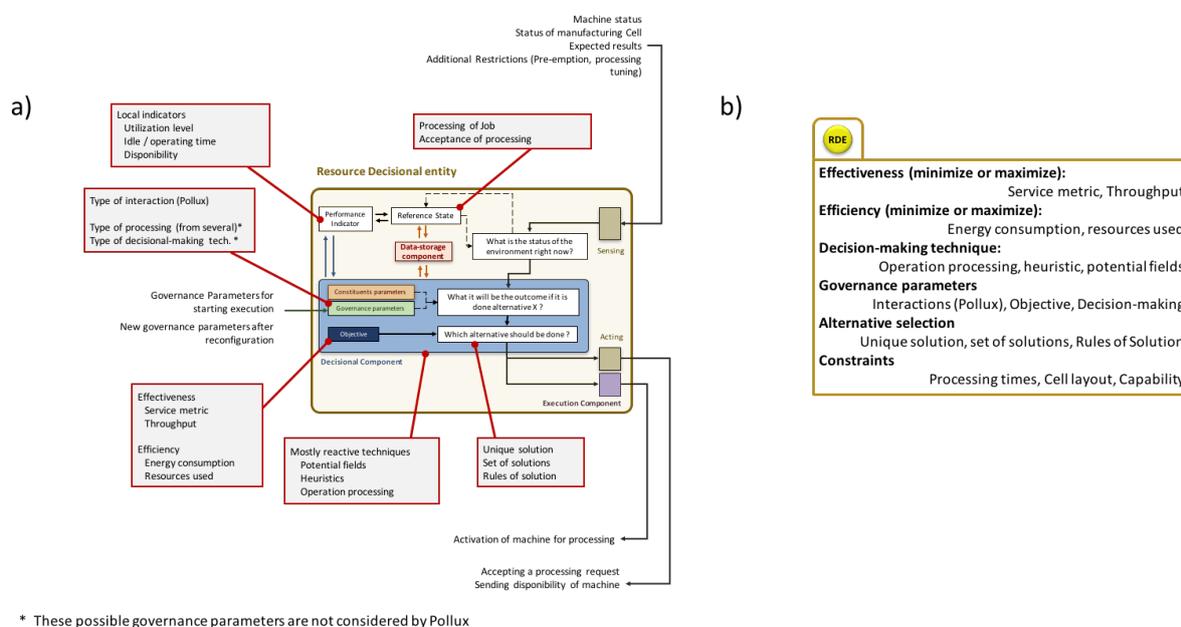
For the decision-making techniques, RDEs have specific protocols that activate the functioning of the physical resource, such as the control of the speed of a conveyor system, the movement direction of a transfer gate or the type of operation and processing time in an assembly machine. Similarly to LDEs, RDEs use two different decision-making techniques depending the values of the governance parameters: default and imposed techniques. While the default techniques imply that the RDEs have protocols that autonomously provide the requested service for their own objectives, imposing techniques follow the actions instructed by the GDE either in a coercive or a liminary interaction.

## Chapter 4. Pollux for a flexible manufacturing system

Each RDE also has a set of governance parameters relating the interaction with the rest of the decisional entities. For the interaction with entities of the coordination level, each RDE has a governance parameter for each GDE. This governance parameter, just as the roles of the LDE regarding interaction with the GDE, can be submissive for coercive interaction with the GDE, or autonomous for liminary and permissive interaction with the GDE. For interaction with other entities on the operation level, each RDE has a governance parameter for interaction with the other entity (i.e. LDE or RDE). This governance parameter, similar to the role of the RDE in relation to the interaction with the same level, can be propositive to offer the completion of an action if another entity accepts, or cooperative in order to carry on a negotiation process with the other entity.

In this example, 21 RDEs are configured on the operation level. The RDEs for the machines range from RDE<sub>1</sub> to RDE<sub>5</sub>, for the material handling devices from RDE<sub>6</sub> to RDE<sub>16</sub>, and for the turn-point transfer gates from RDE<sub>17</sub> to RDE<sub>21</sub>. In this particular case, the RDEs for machines have the objective of processing the requested operation from the job, the RDEs for the material handling device have the objective of transferring the job from the start to the end point at a safe and constant speed, and the RDEs for the turning-point transfer gates have the objective of directing the jobs according to the request.

For the decision-making techniques, while the default technique is configured with the protocol to process the request of the job (i.e. processing, transferring or directing the path), the imposed technique is the protocol for accepting and executing the action imposed by the entities in the coordination level. Note that in this case there is no interaction between the RDEs, as they communicate only with the jobs. The governance parameters of the RDEs correspond to the interaction with the GDE and the RDE. In this particular case, each RDE is configured with a governance parameter with a submissive interaction for the GDE and three governance parameters with a null-relationship for each of the jobs. Figure 30 illustrates the resource decisional entity



**Figure 30** Illustration of resource decisional entity RDE. a) composition, and b) functional diagram

#### 4.4.2.4 Operating mode

One of the main characteristics of the Pollux architecture is the definition of an operating mode. The operating mode gathers the entire set of governance parameters of the decisional entities, which in this particular case focuses on the interactions between the decisional entities. For the starting execution, assuming that the system does not start under a perturbed scenario, the initial and default operation is as it was previously defined in the description of each decisional entity. Nevertheless, as explained in more detail in the following section, the reconfiguration mechanism may change the initial operating mode if required.

The operating mode can be parameterized according to the specific requirements. This may involve the use of arrays, vectors, graphs, matrices or tuples as a representation of the configuration. In this example, the operating mode uses an array containing in an indexed data structure for the whole set of governance parameters. The position of the elements of the array is assigned to a specific governance parameter of a specific decisional entity. The case study is modelled with a single GDE for coordination, 3 LDEs for jobs and 21 RDEs for resources. Therefore, the operating mode for this example contains 24 initial elements (i.e. 3 LDEs and 21 RDEs), 66 subsequent elements (i.e. 1 GDE and 21 RDEs for each of the 3 LDEs), and 84 final elements (i.e. 1 GDE and 3 LDE for each of the 21 RDE). In total, the operating mode for this example is an array of 174 elements that represent the control architecture of the FJSP.

#### 4.2.3 Reconfiguration mechanism

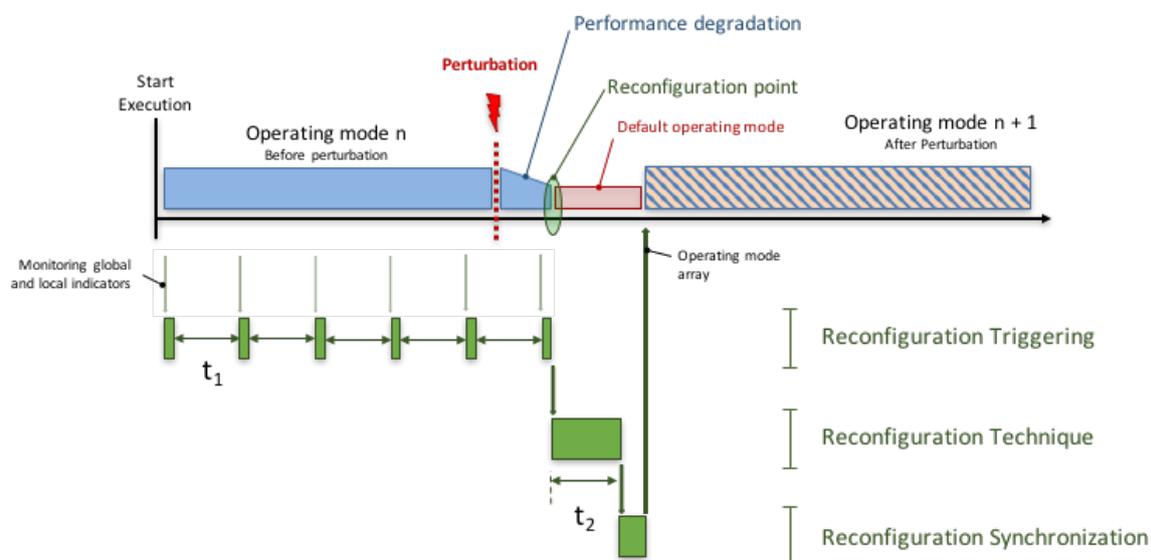


Figure 31 Reconfiguration mechanism functioning over the control system

Figure 31 illustrates the reconfiguration mechanism functioning over the control system. The reconfiguration mechanism of Pollux need to be parameterized according flexible manufacturing system. After the production order is launched according to an Operating mode  $n$ , the reconfiguration mechanism monitors the manufacturing cell through the reconfiguration triggering module. This module may evaluate the manufacturing performance according the performance indicators proposed in subsection 4.3.3. However, this evaluation is executed by the decisional entities. On one side, the GDE evaluates the global indicators, such as the expected makespan at the end of the execution. Some of the techniques for this evaluation might be the evaluation of production trend, an estimation of future events according to current events, exploration of expected behaviours, among many others. On the other side, the LDE and/or RDE evaluates the local indicators, such as unfeasibility to fulfil objectives, occurrence of a perturbation, impossibility of processing the next or subsequent job or operation, among others. Certainly, the infraction of these indicators is measured according the trespassing of a defined threshold for both the global and local indicators. The evaluation of these infractions is executed by the triggering module.

Once the reconfiguration triggering module detects a perturbation, the reconfiguration technique module starts the reconfiguration process. Parallel to this process, Pollux activates a default operating mode in order to react to the perceived perturbation. The default operating mode is the result of changing the governance parameters from the jobs that has been affected by the perturbation and leaving the other jobs with the same governance parameters. This configuration is temporary while the new operating mode  $n + 1$  is applied. During this period, Pollux searches for a new operating mode by running an improvement search process. This process is based in a heuristic, towards-optimal, near-optimal or optimal technique to get the best solution. However, considering that some of the suggested optimization techniques might take a considerable amount of time for its execution, it is recommended to set a limited reconfiguration time. The analysis between the reconfiguration time and the optimization solution is out of the scope of this dissertation. Still, the analysis of satisfactory decision techniques rather than optimization ones is introduced in Jimenez et al. (2017b).

The reconfiguration synchronization module is activated when a new operating mode is found. This module evaluates the possibility that all the jobs can perform its next operation without being blocked. Even though this is a redundancy module as the not possibility of processing would activate a new reconfiguration, the process of this module is included to repair the decision variables to feasible functioning. Some example of this synchronization module is the reparation of the decision variables, or the change to a default governance parameter to the newly affected jobs. An illustration of a functional diagram with possible states on the parameterization of the reconfiguration mechanism is illustrated in figure 32.

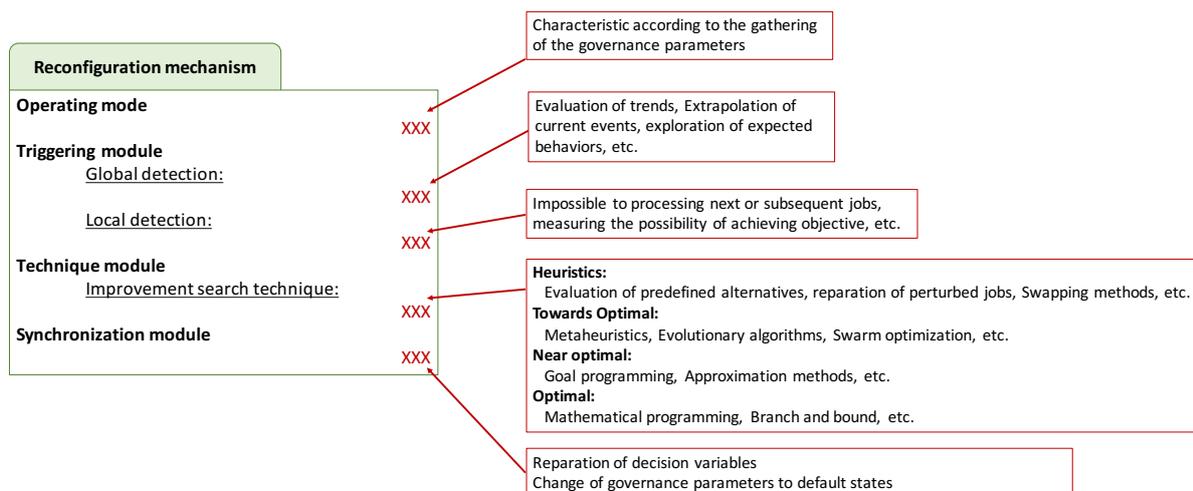
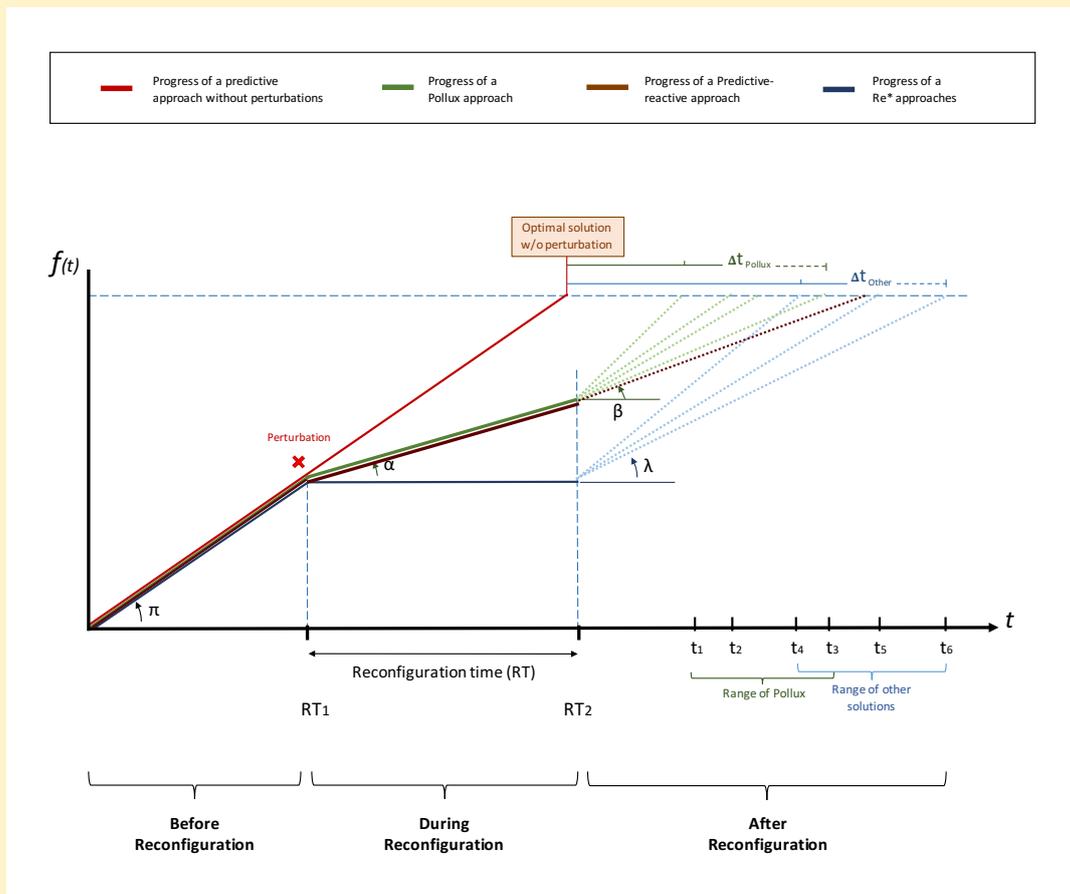


Figure 32 Functional diagram of the reconfiguration mechanism

### Illustration of the contribution of Pollux to the optimality reconfiguration in the manufacturing control.

Following the illustration introduced in Chapter 2, the contribution of Pollux to the optimality of reconfiguration systems in the manufacturing domain is now described. Reconfigurable control systems are centred on the capability of having a flexible control architecture. Pollux, built to make the most of this capability, focuses on using the flexibility not only to customize the control architecture, but also to choose the best alternative for the architecture after the perturbation. The following example explains how the reconfiguration and the flexibility contributes to improve the performance after the reconfiguration and minimize the degradation of the system after the perturbation.

Let us consider the reconfiguration mechanism of a reconfiguration control system when managing a scheduling problem in the manufacturing domain. The decision-making process with the allocation of tasks to resources over a given time period for its processing and its goal is to optimise one or more objectives (Pinedo 2012). The global control problem is to schedule and manage the execution of allocating  $N$  tasks to  $M$  resources, considering that any task can be processed by any resource, pre-emption of tasks is allowed, and the objective is to minimise the total completion time, or makespan. Figure 33 illustrates the execution in terms of the completion progress for different solutions (based on Rojas Ramírez 1999).



**Figure 33** Conceptual model of relation of the progress of completion and the time of execution (Inspired from Rojas-Ramirez (1999)).

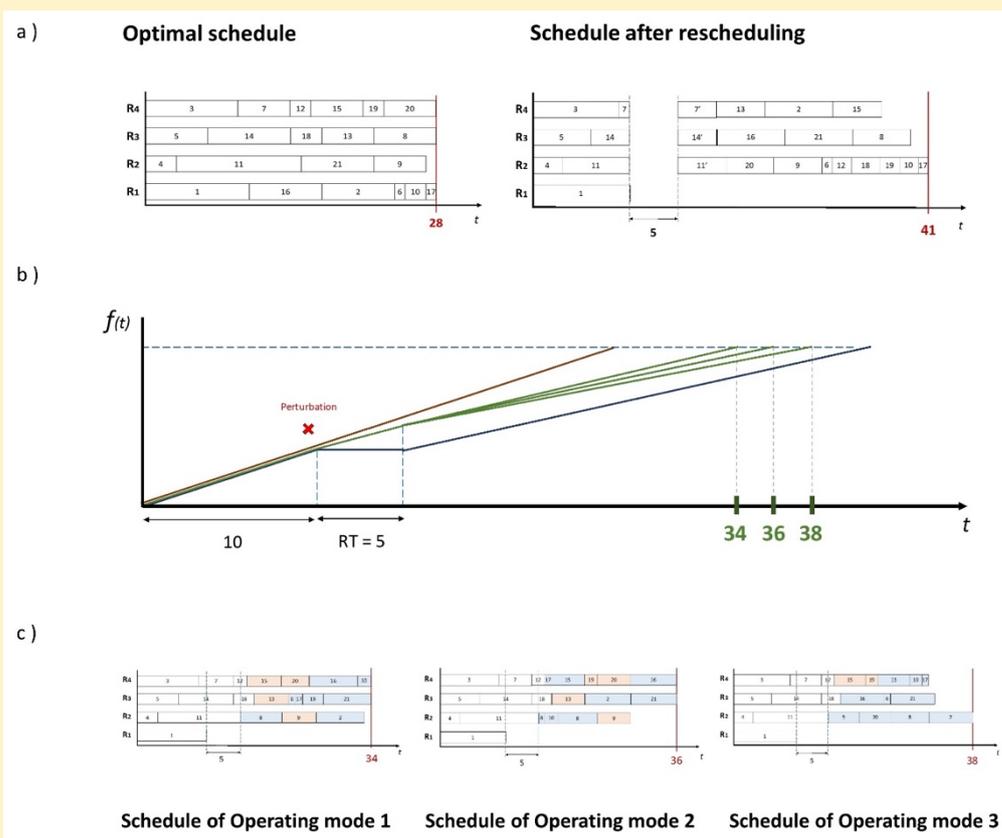
This illustration presents the position of Pollux in respect to other control approaches. It shows the progress of the following control approaches: Pollux’s approach (Green Line); predictive–reactive approaches (Brown line) with a static control architecture; centralized approaches (Blue line) such as rescheduling, retuning or adaptive approaches; and the optimal solution without a perturbation (Red line). For the comparison of these control solutions, three different phases are considered: before the reconfiguration, during the reconfiguration and after the reconfiguration.

**Before the reconfiguration**, it is expected that all the approaches have the same progress following the solution of the predictive technique as under normal conditions.

**During the reconfiguration**, these approaches differ significantly. For Pollux, the reactivity part of the decisional entities works as a default operating mode that intends to respond and progress forward in the completion regardless of the perturbation. The rate of progress decreases from  $\pi$  to  $\alpha$ , but this still partially diminishes the degradation caused by the perturbation. This specific characteristic is not exclusive to Pollux, because any semi-heterarchical control architecture can respond similarly. Centralized approaches

suspend the progress of the execution in order to provide a new configuration (e.g. new schedule, new organization, etc.). In this case, the rate of progress passes from  $\pi$  to  $0$ . These approaches must then seek a minimum time of reconfiguration in order to avoid an extended disruption of the execution progress.

**After the reconfiguration**, these control solutions differ in terms of the results obtained. During the reconfiguration process, Pollux searches for an alternative operating mode to apply after the reconfiguration. Pollux provides different alternatives for the implementation, leading to a range of results (e.g.  $t_1$ ,  $t_2$  and  $t_4$ ). Centralized approaches and predictive-reactive approaches also have a range of results, but these depend on the reconfiguration parameters and level of optimisation featured (e.g.  $t_3$ ,  $t_5$  and  $t_6$ ). Considering the degradation, the difference between the optimal makespan and the chosen alternative, we see that Pollux minimises the degradation by offering on average a shorter  $\Delta t$ . Thus,  $\Delta t_{\text{Pollux}}$  is less than  $t_{\text{Other}}$ . Figure 34 presents an illustration for managing the scheduling of 21 jobs in 4 resource cells. This example assumes the perturbation to be a resource breakdown (Resource 1) which occurred 10 seconds after the beginning of execution.



**Figure 34** Comparison of Pollux with other approaches in an example for the scheduling of 21 jobs into 4 resources.

Figure 34 illustrates the different control solutions in terms of the progress of the manufacturing execution. The top of the figure presents the schedule of the optimal solution in normal conditions and the schedule of a pre-emptive rescheduling approach with centralized architecture (see Figure 34a). The rescheduling result is obtained by running the same model with the uncompleted tasks at the perturbation time. While the optimal solution is a reference to measure the degradation caused by the perturbation, the rescheduling approach is to show a different reconfiguration method that does not process during the reconfiguration but rather searches for an optimal solution after the reconfiguration. It is worth noting that the rescheduling approach obtained the best solution giving different reconfiguration times. In this case, the reconfiguration time was determined to be 5 seconds.

The bottom of the figure shows three schedules obtained from three operating modes. This Pollux application includes a single GDE with the mathematical programming model, 21 LDEs that represent the task to be scheduled, and 4 RDEs that represent the resources (See Figure 34c). It then tested three operating modes in a simulation mode. Each of the operating modes represents a different configuration with a specific degree of governance between the coordination and operation layers of the Pollux architecture. After the reconfiguration time, operating mode 1 maintains 4 tasks with the model instructions (e.g. tasks 9, 13, 15 and 20), operating mode 2 maintains 4 tasks (e.g. tasks 9, 13, 19 and 20), and operating mode 3 maintains 3 tasks (e.g. tasks 15 and 20). For this example, after the reconfiguration, the tasks that follow the GDE instruction intend to execute the processing according to the model, otherwise the resource will process this task as soon it is available. The tasks that do not follow the GDE instructions bid for its execution by a Monte-Carlo simulation, assigning a weight for each task as the inverse of the processing time (e.g.  $1 / \text{processing time of task}$ ). Finally, the middle of the figure (32b) shows the outcome of these approaches and the differences in degradation between the operating modes of Pollux and the rescheduling approach.

In sum, this example highlights two features of Pollux that contribute to the optimality of reconfigurable control systems. First, during the reconfiguration, Pollux runs a default operating mode that takes advantage of the reactivity of the components. It advances in the order of completion by processing the unperturbed tasks and considering the reactivity autonomy of the LDEs when needed. Second, an explanation is presented regarding how the range of operating modes after the reconfiguration allows us to have better controllability of the control system and makes it possible to run an optimisation process to choose the most adequate (e.g. optimal) operating mode.

### 4.5 Summary

This chapter presents an implementation of the proposed approach, Pollux, into the manufacturing domain. This implementation included Pollux as an extended MES of an ERP architecture between the enterprise planning (level 4) and physical manufacturing operations (Level

0). Then, Pollux is responsible for managing the planning, scheduling and control of a manufacturing operations. For this implementation, this dissertation considers in specific a FJSP to parameterize the proposed approach. This example problem within the FMS is chosen as it eases the implementation by featuring the sufficient flexibility that need to be managed by a reconfiguration system, allowing the inherence during the operation execution, and the optimization of the dynamic configuration is gaining interest in a wide range of fields in the ICT community. Although, this chapter applies Pollux for the manufacturing field, the flexibility, customizability and the optimization reached apply to the improvement of the efficiency, effectiveness and reactivity of any fields where a discrete-event operation is executed (e.g. logistics, aeronautics, health-care, etc.).

Initially, for the stating the problem to resolve in the manufacturing domain, the description of the FJSP is detailed. The problem formulation and the constraints related to the jobs and manufacturing cell are presented. We remark the decision-making of this FJSP, explaining the flexibility given for the completion of the production order (e.g. release sequence of jobs, machine sequence of jobs and routing path of jobs). Afterwards, some performances indicators in the manufacturing domain are described. Here, it is presented the indicators according to the need in the local and global indicators within the manufacturing systems.

Afterwards, the implementation of Pollux to the manufacturing domain is described. As Pollux is created as a reference architecture to host different techniques in the predictive, reactive and reconfiguration level. This chapter proposes potential techniques to include to the proposed approach. The purpose of showing the openness in the configuration is to highlight the wide flexibility featured by Pollux. During the described implementation, it is presented each model that was used to construct the architecture. The decisional entity model is implemented to represent that jobs and machines from the manufacturing cell. These are responsible of the manufacturing execution. The structure and behaviour characteristics define together the operating mode within the control solution. Considering this, the operating mode and the interaction between the GDE, LDE and RDE describe the control architecture of the extended MES. For the reconfiguration mechanism, it is explained the dynamism with the control architecture and the impact with the manufacturing system. Finally, it is included an illustration of the contribution for the optimal reconfiguration of Pollux in the manufacturing domain.

Considering that chapter 4 proposes how could be implemented of Pollux to the manufacturing domain, the next chapter implements in detail a specific FMS from a manufacturing cell located in Valenciennes, France. It is also conducted two simulated and one real experimental studies in the implemented FMS in order to evaluate the benefits of our approach. The results of these experiments and the benefits of including optimality-based principles to a reconfiguration control system is reported.

# Chapter 5

## **POLLUX-AIP: An experimental case study**

### **5.1 Introduction**

Considering that previous chapter presented the implementation of Pollux in a general FMS, this chapter implements and validates the benefits of the proposed approach conducting three experiments in a simulated and real scenario of a real FMS located in Valenciennes (France). While the previous chapter presented the implementation of Pollux in a general FMS, this chapter aims to implement and validate the benefits of the proposed approach by conducting three experiments in a simulated and real scenario of a real FMS located in Valenciennes (France). Section 2 details the data related to the AIP cell, and section 3 describes a general implementation of Pollux to the detailed AIP cell. The validation of the proposed approach is conducted in three experiments described in section 4. In this section, the first experiment (Experiment A) assesses the feasibility of the different operating modes and the effectiveness of Pollux in a simulated scenario programmed in an agent-based software called Netlogo. The second experiment (Experiment B) conducts the execution in the real AIP cell to demonstrate the applicability of Pollux in a real scenario and to reinforce the improvement achieved in the efficiency, effectiveness and reactivity. Finally, the third experiment (Experiment C) implements a sustainable version of Pollux in a simulated Netlogo model in order to present the feature of multi-criteria objectives, show an energy-aware approach, and demonstrate the controllability and optimisation reached.

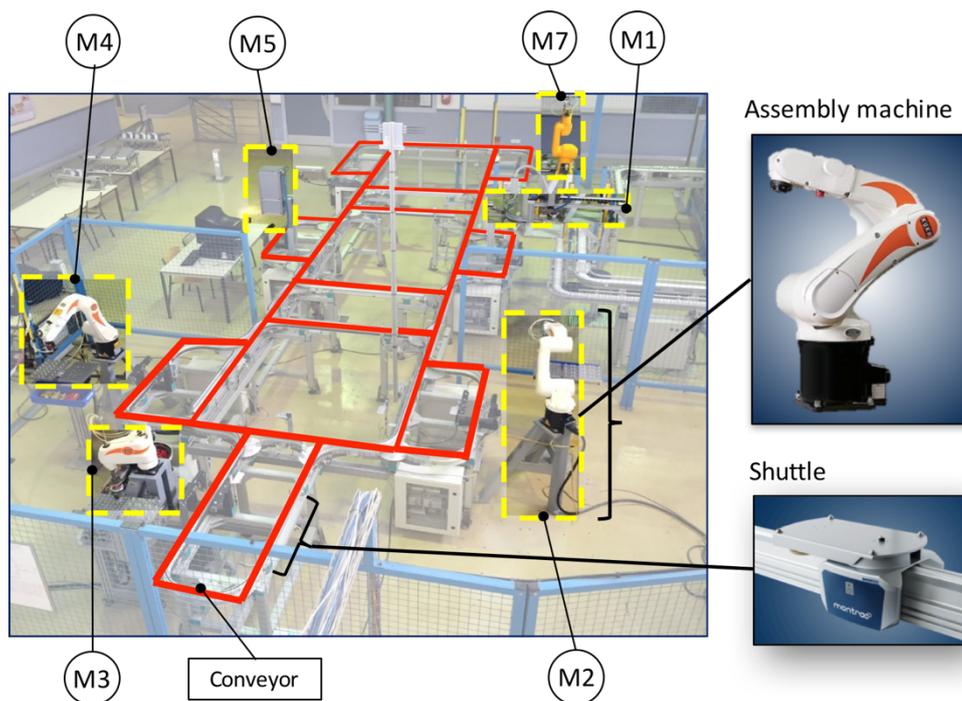
### **5.2 Valenciennes AIP-PRIMECA: a flexible manufacturing system**

This section describes the use of a case study used to assess the proposed approach. First, this section introduces the specifications of a manufacturing cell. Then, in order to formalize the problem under consideration, the mathematical formulation of the flexible job shop problem is presented as it relates to this manufacturing cell. The problem, which is described in Trentesaux et al. (2013), is based

on a flexible manufacturing system located at the University of Valenciennes, France (see Figure 35). The mathematical formulation of the FJSP is inspired by the formulation in Demir and İşleyen (2013). This information is used in the following section, where Pollux is applied to the flexible manufacturing cell.

### 5.2.1 Description of the manufacturing cell

The flexible manufacturing system referenced in this chapter is an assembly cell located at the Université de Valenciennes et du Hainaut-Cambrésis (France), called AIP-PRIMECA Valenciennes (*Atelier Inter-établissement de Productique (Pôle de Ressources Informatique pour la Mécanique)*). The AIP cell (short for AIP-PRIMECA) is an assembly of industrial components, including robots, sensors, actuators, and a handling-system, that are positioned together to process a set of defined jobs (See Figure 35). This cell is an example of a flexible manufacturing system that features the following types of flexibility: machine flexibility, as various operations can be performed without a set-up change; routing flexibility, as there are different transfer routes between at least two different machines; process flexibility, as a set of products can be manufactured without any major setup changes (ElMaraghy 2006); and machine-sequence flexibility, as alternative machine sequences can be followed to produce an unique part (Baykasoğlu & Özbakır 2008).



**Figure 35** AIP-PRIMECA: Flexible manufacturing system located in University of Valenciennes

After this general description, the AIP specifications are depicted in detail. This information is needed for both the application of Pollux for the AIP primeca and the implementations in the experimental studies. The detailed description is divided between the layout of the AIP system, which includes the description of the machines and the material handling system, and the production specifications, which includes the products, components and the manufacturing process.

### **Layout of the AIP cell.**

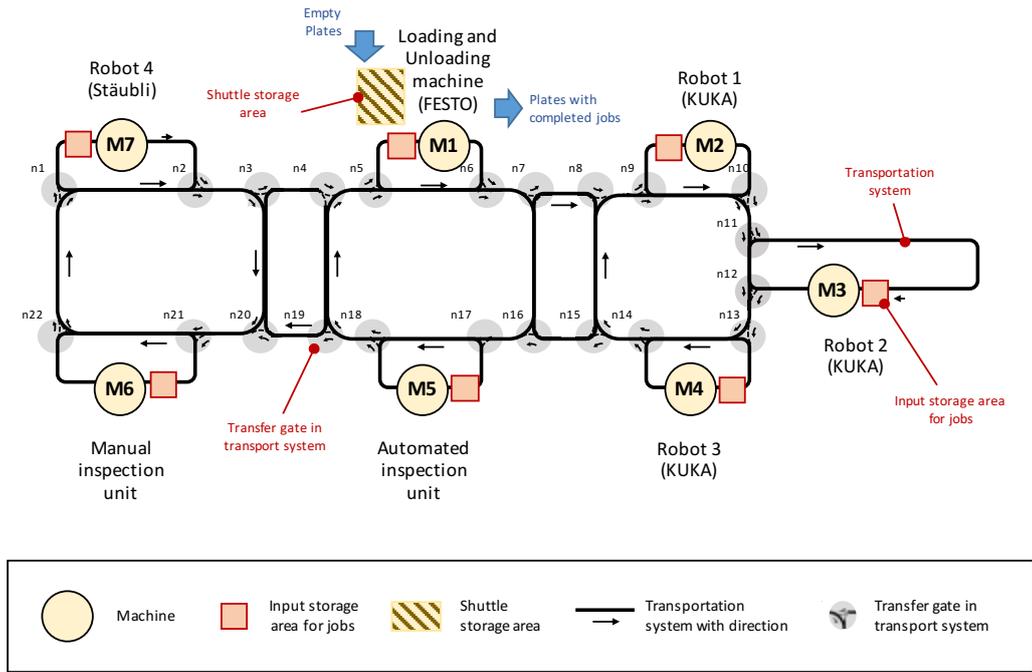
the AIP manufacturing cell is composed of seven machines connected through a unidirectional and flexible transport system. This cell can process specific jobs, which are transferred in self-propelled shuttles through the transport system. An illustration of the AIP cell is presented in figure 33.

**Machines:** There are seven machines in the AIP cell. These are responsible for executing the manufacturing operations (i.e. each job is composed of a set of operations) of the production system. The AIP cell features partial flexibility because a subset of machines is able to perform the same manufacturing operations, and some operations are performed only by a single machine. Each machine has an input storage area that serves a supply position of the machine. The machines available in the cell are:

- a loading / unloading machine unit (M1)
- three assembly robots KUKA (M2, M3 and M4)
- an automatic inspection unit (M5)
- a manual inspection unit (M6)
- a redundant assembly robot STÄUBLI (M7)

**Material handling system:** The system responsible for handling the jobs within the AIP cell is a monorail. This transport system, specifically a Montech system (Montech International, 2017), leads the shuttles through the cell and guides them to their destinations by rotating transfer gates distributed within the transport system. The transport system contains 29 nodes: 7 nodes for the machines (from M1 to M7), and 22 nodes for the transfer gates (from n1 to n22). It can be considered a strong connected graph. The nodes of the transfer gates may be either divergent or convergent. A divergent node is a point in the transport system where a route must be chosen from two different alternatives. A convergent node is a point of the monorail where two different routes are joined together to continue through a same route. In addition, the AIP cell also contains a shuttle storage area for the loading of jobs to the AIP cell, an input storage area as a buffer area for each machine and the shuttles that are transported through the monorail. The layout of the AIP cell is illustrated in figure 36.

### **Production specification of the AIP cell**



**Figure 36** Layout of the AIP-PRIMECA flexible manufacturing cell, Valenciennes.

**Products or jobs:** seven types of jobs are offered to clients. Each of these products is a corresponding job to be processed through the AIP cell. These jobs are assigned the letters “B”, “E”, “L”, “T”, “A”, “I”, and “P”. The jobs are built through an assembly of a set of components. These components need to follow a subset of operations, executed by the machines, to complete a particular job. Figure 37 illustrates the jobs that can be processed (types of jobs) in the AIP cell.

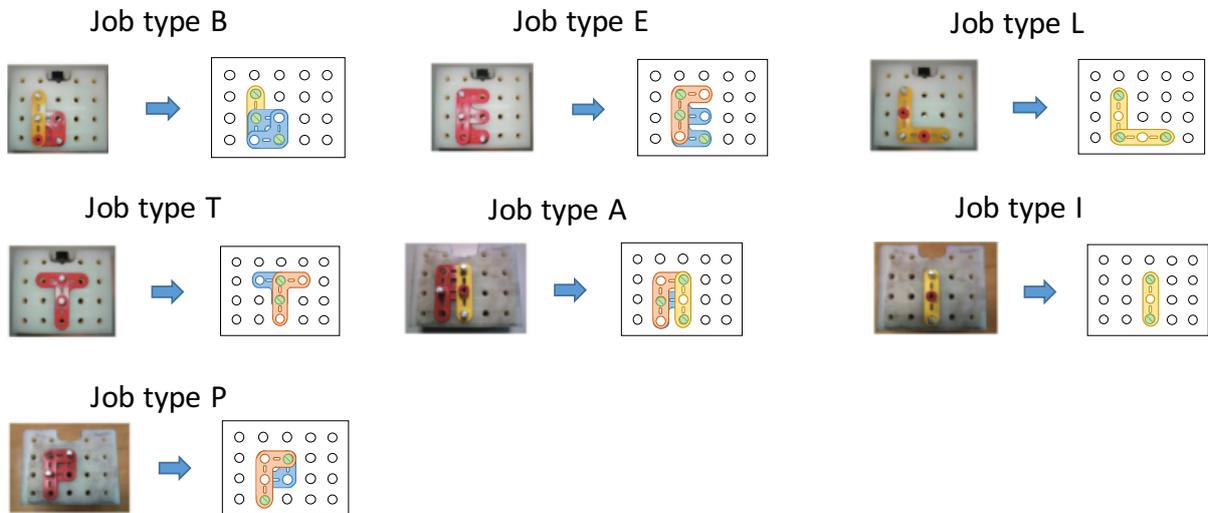


Figure 37 Types of job processed by the AIP cell

**Components:** six components are needed for the processing of the job: “Plate”, “Axis component”, “I component”, “L component”, “r component” and “screw component”. These components are the input material for the AIP cell and it is assumed that the AIP cell has an infinite capacity of this components. The components and bill of materials are illustrated in Figure 38.

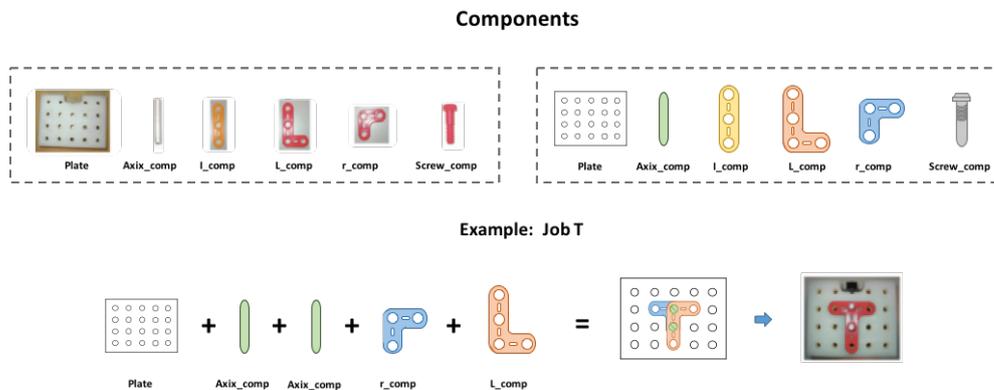


Figure 38 Components of the AIP cell

**Operations:** eight manufacturing operations can be processed in the AIP cell: “Plate loading”, “Axis mounting”, “r\_component mounting”, “I\_component mounting”, “L\_component mounting”, “Screw mounting”, “Inspection”, and “Plate unloading”. For example, the operation “r\_component mounting” means that the “r component” is mounted on the plate. Each of these operations can be processed by at least one machine of the AIP cell. Figure 39b shows the machines that can execute each operation and the processing times required by each machine. The spaces with dots in the table mean that this machine cannot process that operation.

## Conclusions and Future work

**Production sequence:** each job must execute a specific production sequence for completion. In general terms, the jobs execute the following production sequence: a single load, a series of component mountings, a single inspection and a single unloading. The detailed production sequence is a subset of the operations. Figure 39a shows the operations and the production sequence of the jobs. Between two successive manufacturing operations, transportation may be required between two machines if either of the two operations are not conducted at the same place or if so instructed for other reasons. The transportation times are presented in figure 39c.

a)

Job	Production sequence
<b>B</b>	O7-O1-O1-O1-O2-O2-O5-O3-O6-O8
<b>E</b>	O7-O1-O1-O1-O2-O2-O4-O6-O8
<b>L</b>	O7-O1-O1-O1-O5-O5-O3-O3-O6-O8
<b>T</b>	O7-O1-O1-O2-O4-O6-O8
<b>A</b>	O7-O1-O1-O1-O2-O4-O5-O3-O6-O8
<b>I</b>	O7-O1-O1-O5-O3-O6-O8
<b>P</b>	O7-O1-O1-O2-O4-O6-O8

b)

	Processing times (Seconds)							
	M1	M2	M3	M4	M5	M6	M7	
Axis mounting	O1	·	20	20	·	·	·	20
r_component mounting	O2	·	20	20	·	·	·	20
l_component mounting	O3	·	·	·	20	·	·	20
L_component mounting	O4	·	20	·	20	·	·	20
Screw mounting	O5	·	·	20	20	·	·	·
Inspection	O6	·	·	·	·	5	·	·
Plate loading	O7	10	·	·	·	·	·	·
Plate unloading	O8	10	·	·	·	·	·	·

c)

Source nodes	Destination nodes																	
	n2	n4	n6	n8	n10	n12	n14	n16	n18	n20	n22	M1	M2	M3	M4	M5	M6	M7
n2	·	4	·	·	·	·	·	·	·	5	·	·	·	·	·	·	·	·
n4	·	·	4	·	·	·	·	·	·	·	·	5	·	·	·	·	·	·
n6	·	·	·	4	·	·	·	5	·	·	·	·	·	·	·	·	·	·
n8	·	·	·	·	4	·	·	·	·	·	·	5	·	·	·	·	·	·
n10	·	·	·	·	·	3	·	·	·	·	·	·	·	11	·	·	·	·
n12	·	·	·	·	·	·	4	·	·	·	·	·	·	·	5	·	·	·
n14	·	·	·	5	·	·	·	4	·	·	·	·	·	·	·	·	·	·
n16	·	·	·	·	·	·	·	·	4	·	·	·	·	·	·	5	·	·
n18	·	5	·	·	·	·	·	·	·	4	·	·	·	·	·	·	·	·
n20	·	·	·	·	·	·	·	·	·	·	4	·	·	·	·	·	7	·
n22	9	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	10
M1	·	·	·	6	·	·	·	7	·	·	·	·	·	·	·	·	·	·
M2	·	·	·	·	·	5	·	·	·	·	·	·	·	13	·	·	·	·
M3	·	·	·	·	·	·	6	·	·	·	·	·	·	·	7	·	·	·
M4	·	·	·	7	·	·	·	6	·	·	·	·	·	·	·	·	·	·
M5	·	7	·	·	·	·	·	·	·	6	·	·	·	·	·	·	·	·
M6	12	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	13
M7	·	6	·	·	·	·	·	·	·	7	·	·	·	·	·	·	·	·

**Figure 39** Information of the AIP cell, a) Production sequence per jobs, b) processing time component, and c) transport time between machines.

### 5.2.2 Mathematical formulation of the Flexible manufacturing cell

The mathematical formulation of the FSJP is presented below. This subsection describes the variables, parameters and constraints related to this problem to formalize the present in the FMS. This formalization is used to position the components in the Pollux architecture. A detailed description of this formulation is found in Trentesaux, et al. (2013)

### Sets and indices

- J : set of jobs  $\{1, \dots, j, \dots, N\}$ ,
- M : set of machines  $\{1, \dots, m, \dots, M\}$ ,
- $O_j$  : set of sequenced operations  $i$  of job  $j$   $\{O_{1j}, \dots, O_{ij}, \dots, |O_j|\}$
- $R_{ij}$  : set of machines  $m$  that can process operation  $O_{ij}$

### Parameters

- $p_{ijk}$  : processing time of operation  $O_{ij}$  in machine  $k$
- $tt_{mm'}$  : transfer time from machine  $m$  to machine  $m'$
- $W$  : maximum number of jobs simultaneously in the shop floor
- $cb_m$  : capacity of input buffer for machine  $m$
- $\varepsilon$  : Small number
- $\beta$  : large number

### Variables

- $t_{ij}$  : Completion time of operation  $O_{ij}$
- $u_{ijm}$  :  $\begin{cases} 1, & \text{if operation } O_{ij} \text{ is processed in machine } m \\ 0, & \text{otherwise} \end{cases}$
- $b_{iji'j'}$  :  $\begin{cases} 1, & \text{if operation } O_{ij} \text{ is processed before the operation } O_{i'j'} \\ 0, & \text{otherwise} \end{cases}$
- $tm_{ijmm'}$  :  $\begin{cases} 1, & \text{if job } j \text{ is transferred from } m \text{ to } m' \text{ after processing operation } O_{ij} \\ 0, & \text{otherwise} \end{cases}$
- $w_{ijm}$  : waiting time of operation  $O_{ij}$  in the input buffer of machine  $m$
- $wv_{iji'j'}$  :  $\begin{cases} 1, & \text{if operation } O_{ij} \text{ is waiting in the input buffer of machine } r \text{ that } O_{i'j'} \text{ is processed} \\ 0, & \text{otherwise} \end{cases}$

### Constraints

$$t_{ij} + p_{i+1jk} + w_{i+1jm} + \sum_{\forall m \in R_{ij}} tt_{mm'} tm_{ijmm'} \leq t_{i+1j} \quad \forall i, j \in O_j; \forall m' \in R_{ij} \quad (1)$$

$$\sum_{\substack{\forall m, m' \in R_{ij} \\ m \neq m'}} tm_{ijmm'} \leq 1 \quad \forall i, j \in O_j \quad (2)$$

$$t_{ij} + p_{i'j'm} u_{i'j'm} + \beta b_{iji'j'} \leq t_{i'j'} + \beta \quad \forall i, j, \in O_j; \forall i', j', \in O_{j'}; \forall m \in R_{ij} \quad (3)$$

$$b_{iji'j'} + b_{i'j'ij} \leq 1 \quad \forall i, j, \in O_j; \forall i', j', \in O_{j'} \quad (4)$$

$$\sum_{\forall m \in R_{ij}} u_{i'j'm} = 1 \quad \forall i, j, \in O_j \quad (5)$$

$$u_{ijm} + u_{(i+1)jm'} - 1 \leq tm_{ijmm'} \quad \forall i, j, \in O_j \forall m, m' \in R_{ij}, m \neq m' \quad (6)$$

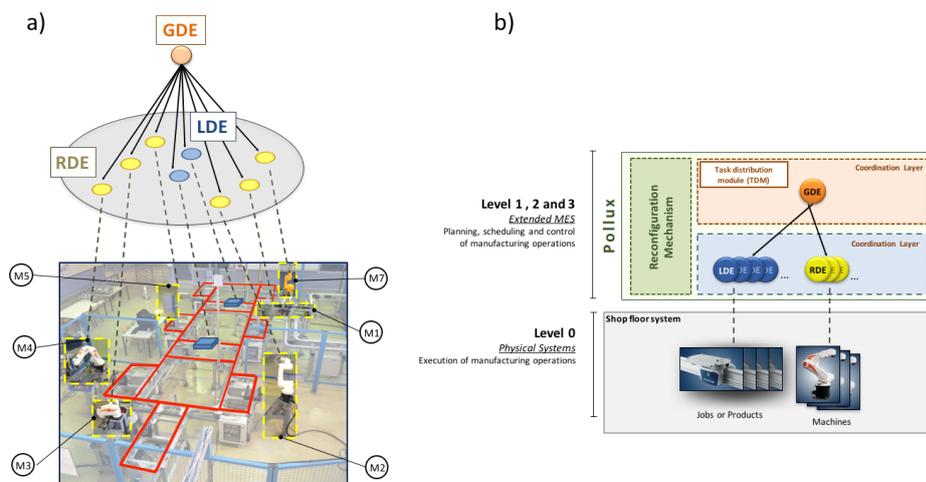
$$u_{ijm} + u_{(i+1)jm'} \geq (1 + \varepsilon)tm_{ijmm'} \quad \forall i, j, \in O_j \forall m, m' \in R_{ij}, m \neq m' \quad (7)$$

### Objective Function

$$\min Z = \max_{\forall i \in O_j, \forall j \in J} t_{ij} \quad (8)$$

The FSJP case is described in this mathematical formulation. This formulation resolves the planning and scheduling of the jobs during execution. This resolution is used to set the predictive solution to the manufacturing operations. The objective function (eq. 8) minimize the completion time of the last operation of the last processed job. The decision variables set the completion time  $t_{ij}$ , which solve the machine sequence in the problem; the allocation of operations to the machine  $u_{ijm}$ , which set the release sequence to the manufacturing cell (e.g. specifically the uploading machine M1); and the transportation between machines  $tm_{ijmm'}$ , which solve the routing path of jobs within the conveyor. In addition, some auxiliary decision variables are created in order to model the queuing and the looping within the transportation system, such as order of processing a pairwise operations, the waiting time of jobs in queue and the binary variable that say if it has wait in queue. The constraints are: the precedence constraints, which ensure the order of processing of operations, including the completion time of the previous operations, the waiting time and the transportation time (eq. 1). The transportation constraints, which ensure that a job has a unique path between two machines (eq. 2). The disjunctive constraints, which ensure that a machine processes one operation at time, and the operation is processes only once (eq. 3, 4 and 5). And Finally, the allocation/transportation constraints, which ensure the relation of time between the moment where it is allocated the jobs and the transportation of them. In the section that follows is explained how the components of Pollux are modelling each of the elements of this FJSP.

### 5.3 Pollux implementation to the AIP cell (Pollux-AIP)



**Figure 40** Pollux implementation to the AIP Primeca, a) relation virtual-physical system, and b) Pollux architecture for FMS

### 5.3.1 External view: Purpose and environment

As explained in the previous chapter, the global control problem for a flexible manufacturing system consists of scheduling and completing the execution of a production order. The **purpose** of this implementation is to minimise the makespan of the production order, whether or not it is subject to a perturbation event. The decision capability of the control system is derived from the FJSP. Therefore, the control system must manage the following decisions: a) controlling the release sequence of jobs into the AIP cell, b) controlling the machine sequence of jobs to complete the operations of each job, and c) controlling the routing path of jobs to guide the jobs through the material handling system of the AIP cell. In the section that follows, Pollux is implemented, defining both the control architecture and the reconfigurable mechanism. Considering that Pollux is a system to plan, schedule and control manufacturing operations (See Figure 24, Chapter 4), the environmental interactions are limited by the enterprise system (level 4), which assigns the execution of the production order to Pollux, and the physical system (Level 0), which is the controlled system that performs (in the real AIP cell) the manufacturing execution.

### 5.3.2 Internal view: control architecture and reconfigurable mechanism

The **control architecture** of this implementation is organised in three layers: the coordination layer, operation layer and physical layer (see Figure 40). While the coordination level and the operation level host the decisional entities, the physical layer hosts the controlled system which performs the execution. The resources and jobs of the flexible job shop are thus located in the physical layer. For the implementation, three types of decisional entities are used: local decisional entities (LDEs), resource decisional entities (RDEs), and global decisional entities (GDEs). The LDEs, located in the operation layer, are responsible for coordinating the online scheduling and guiding the jobs located in the physical layer (raw materials, work-in-progress or finished products). The LDE contains all the information related to the manufacturing of the jobs, such as the bill of materials, the production sequences and the constituent operations. The main objectives are to perform the operation and support the reactivity requirement in the event of a perturbation. The RDEs, located in the operation layer, are responsible for controlling the service-oriented resources located in the physical layer (conveyors, robots, storage systems, AGV, etc.), and fulfilling the objective assigned to the flexible job shop (i.e. product processing, energy management, machine productive/idle management, maintenance management). These entities include all the information related to the resource, such as processing times, layout information, storage capacity and operation capacity. The main objective of the RDEs is to maintain resource-related goals whilst ensuring job processing. The GDEs, located in the coordination layer, are responsible for the offline scheduling and for fulfilling the global objectives (e.g. completion of production order or energy management). The decisional entities and the corresponding governance parameters are detailed in each experiment in section 4 of this chapter.

The **reconfiguration mechanism** holds the three modules introduced in chapter 4. The reconfiguration triggering module relies on detecting perturbations in the FJSP by evaluating global and local indicators. On the one hand, the local indicator, which is located in the LDE, measures performance in the completion of a local objective. This indicator is modelled with the function  $f(t, X)$  and contains a lower  $\alpha_1$  and an upper  $\beta_1$  threshold, violation of which will activate the reconfiguration. On the other hand, the global indicator measures the performance in completion of the global objective. This indicator is  $g(t, X, S)$  and also contains a lower  $\alpha_2$  and an upper  $\beta_2$  threshold. The reconfiguration

technique module relies on choosing an operating mode from the set of alternatives. This component considers any function as a mathematical or a simulation-based function, which is explained in each experiment. The reconfiguration synchronisation modules implement the new operating mode and resume the execution.

### 5.4 Experimental protocol

This section presents the experimental protocol to evaluate the proposed approach. The intention of conducting these experiments is to evaluate Pollux, assess the improvement gain in efficiency, effectiveness and reactivity over other known solutions and test the level of recovery and stability provided. The protocol is carried out in three types of experiments.

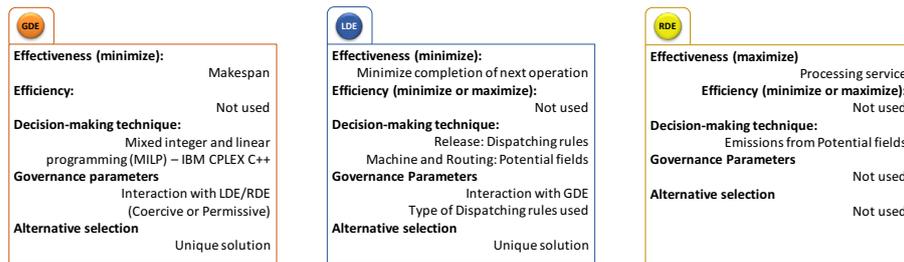
- **Experiment A** uses a simulation model to evaluate the diversity of results yielded by the operating mode of Pollux after reconfiguration, compare the recovery of the makespan among five different approaches, and present a detailed analysis of a reconfiguration event for Pollux and another solution approach.
- **Experiment B** is conducted in the real AIP cell, where the reconfigurable mechanism of Pollux is parametrized with a parallel genetic algorithm, to support the reconfiguration during manufacturing execution.
- **Experiment C** uses a NETLOGO model to test a simulated and energy-aware scenario of Pollux, including an efficiency metric for energy consumption.

#### 5.4.1 Experiment A.

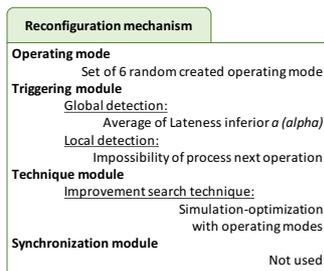
**Pollux application:** For Experiment A, Pollux is implemented following the general description presented in chapter 3. Figure 41 presents the reconfigurable control architecture (i.e. the control architecture and reconfiguration mechanism) and the definition of the decisional entities. Further information about this experiment is found in Jimenez, Bekrar, Trentesaux & Leitao (2016a)

# Conclusions and Future work

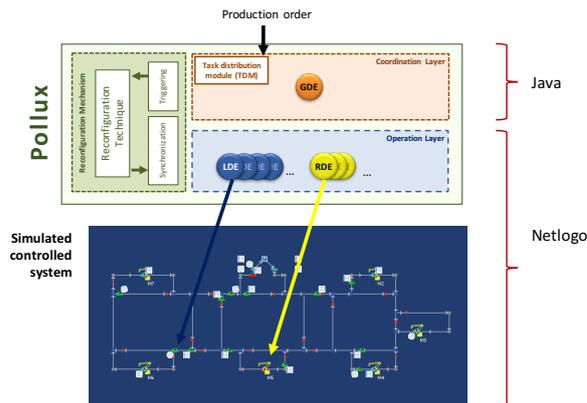
## a) Decision entities for Experiment A



## b) Reconfiguration mechanism for Experiment A



## c) Architecture for Experiment A



\* Each entity represents an object in the AIP cell. This figure do nor relation all objects

## Control architecture.

Figure 41 Implementation of Pollux for experiment A

- Coordination layer:** a single GDE contains a mixed integer linear programme (MILP) for solving the release sequence, machine sequence and routing path of jobs 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 release sequence, machine sequence and routing path of jobs. The possible roles are either coercive or permissive.
- Operation layer:** as many LDEs as jobs in the production order are created. Each LDE contains two decisional techniques for releasing, machine sequence and routing path of jobs. 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 release sequence, the possible values for the governance parameter are (a) dispatching rule 1: to adopt the decision-making technique as the longest processing time (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 and routing of jobs. A potential fields approach is used for machine 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 is to emit the potential fields when it is needed.

- **Operating modes:** the operating modes of Pollux surge 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:

$$\text{Operating mode}1 = \{Coercive, Coercive, Coercive, Dispatching Rule1; Potential fields; \dots\}$$

where the first three elements of the vector are the governance parameters of the GDE for dispatching the machine sequence and routing path of the jobs, and all the other elements are the governance parameters arranged in pairs according to the number of LDEs in the production order. In experiment A, the general fitness for each operating mode, which will provide the quality indicator for the reconfiguration process, is calculated based on a simulation model of the AIP-PRIMECA developed using the Netlogo software. In this Experiment, considering that all LDEs may have the same governance parameters, six operating modes are randomly created to evaluate the reconfiguration. Nevertheless, the diversity of the operating modes created emerges from the variability of the governance parameters of the GDE and the LDEs.

### Reconfiguration mechanism

- **Reconfiguration triggering module:** the reconfiguration triggering module defines two protocols to detect perturbations and consequently trigger a reconfiguration. In the first, 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:

$$f(t, X) = \frac{\sum_{v_j} (b_j - t)}{n_j} \geq \alpha$$

where  $b_j$  is the expected completion time of the next operation derived from the schedule from the GDE,  $n_j$  is the number of jobs in the system,  $t$  is the current simulation time,  $X$  is the current state of the shop floor, and  $\alpha$  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 reconfiguration is activated. It is recommended to set the  $\alpha$  parameter to a negative value in order to permit degradation in the system up to a considerable limit. For the experimental case study, the parameter  $\alpha$  is set to  $-10$  for global detection. The local detection mechanism is responsible for detecting machine breakdowns. For this experiment, local triggering is initiated by the LDE, which informs the reconfiguration mechanism that it is unable to achieve its objectives. In this case, once the information is received, the reconfiguration 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 reconfigure is communicated.

- **Reconfiguration technique module:** the reconfiguration technique module is implemented on the basis of an improvement search technique. When it is reconfigured, this mechanism gathers

## Conclusions and Future work

information about the CA 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 execution time for the order of seven jobs is 0.786 s on average. Thus, a reconfiguration 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.

- **Reconfiguration synchronisation module:** because it is evaluating only six operating modes, the reconfiguration synchronisation module is not used in this case study. The governance parameters do not specifically change the intentions of the jobs (decisions), but they do dictate the behaviour of the entities. Thus, when the configuration is altered, the entities function normally and define the intentions of the jobs when the new operating mode is applied. In this case, synchronisation is not needed.

### Experimental protocol

**Experiment purpose:** The main objective of experiment A is to evaluate the inclusion of the reconfigurable mechanism in terms of assessing the efficiency, effectiveness and reactivity of the proposed model compared with other architectures. The case study presented in this section is carried out to evaluate the proposed approach in a simulated scenario. To this end, these simulations are conducted in an agent-based simulation software, NETLOGO (Wilensky 1999), which models the AIP cell located in Valenciennes, France. These simulations are two-fold. First, several simulations are conducted to corroborate the diversity of the results yielded by several operating modes during the execution. Second, the performance of five different control solutions after perturbation is compared, and a detailed comparative analysis between Pollux and a particular architecture is presented. This experimental approach was adopted primarily to ensure that each operating mode contains a particular control strategy, and it certainly helps to set a criterion to choose the best alternative and it is thus valuable in optimising the reconfiguration process.

**Experiment method:** the simulation test considers three variables: the data-set, which describes the production order to process; the disruption type; and the time of the perturbation. This encompasses three factors with 3, 2 and 3 levels, respectively. Table 5 present the levels for each factor in experiment A.

**Table 6** Level for each factor of experiment

Factors	Levels		
<b>Data Set</b>	<b>A0</b> 7 jobs (B, E, L, T, A, I, P) at the same time	<b>B0</b> 7 jobs (B, E, L, T, A, I, P) considering only five jobs the same time	<b>C0</b> 7 jobs (B, E, L, T, A, I, P) considering only five jobs at the same time
<b>Disruption type</b>	DisR1 M3 Breakdown and it cannot process further operations	DisR2 M3 increase operations time by 100%	-----
<b>Moment of perturbation</b>	22 time units	195 time units	239 time units

## Conclusions and Future work

The possible states of the data-sets are A0, B0 and C0, and belong to the AIP PRIMECA benchmark (Trentesaux et al. 2013). Data-set A0 processes seven jobs (B, E, L, T, A, I, P); data-set B0 processes seven jobs (B, E, L, T, A, I, P) considering only five jobs at the same time; and data-set C0 processes seven jobs (B, E, L, T, A, I, P), also considering only five jobs at the same time. 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 continues until all the jobs are processed. Considering the combinations of the factor's levels, this experiment has 18 different scenarios. Finally, the objective of the experiment was to compare the makespan of all the scenarios. Thus, the experiments test the proposed approach of Pollux and four different control systems with reconfiguration characteristics, named CS1, CS2, CS3, CS4 and CS5 (See Table 6). While CS1 is the approach proposed in this thesis (Pollux), CS2 and CS3 are based on the ORCA architecture (Pach et al. 2014), which is included in the reviewed literature. This architecture is chosen because we believe that it is the closest to our proposal. The difference between CS2 and CS3 is that for CS2 all jobs reconfigure their behaviour from a predictive scheduling technique to a potential field's approach, while for CS3 only perturbed jobs reconfigure their behaviour from a predictive scheduling technique to a potential fields approach. CS4 is an S-CA 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 in the table below:

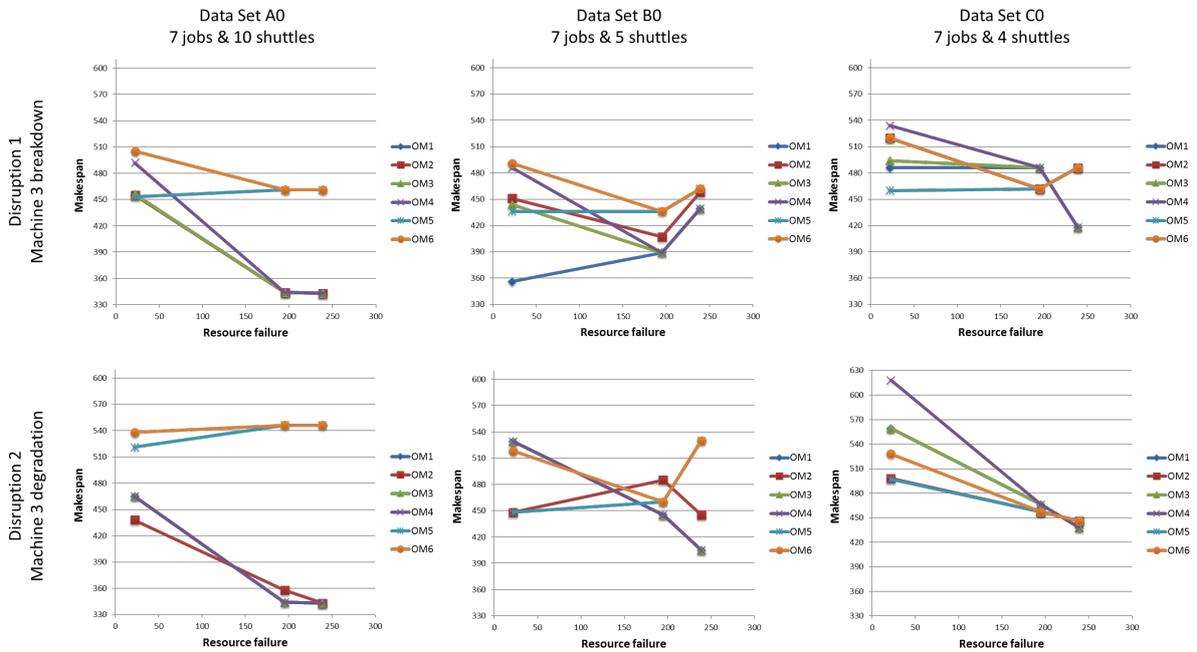
**Table 7** Description of the proposed control systems

<b>Approaches</b>	<b>Description</b>
<i>Control system 1 CS1 (Pollux)</i>	Proposed approach with a reconfiguration mechanism that steers the operating mode. While the GDE is configured with a MILP mathematical program the LDEs and RDEs are configured with a shortest processing time (SPT) win for the release sequence of the jobs and potential fields for the machine sequence and routing path decisions.
<i>Control system 2 CS2 (Pach et al. 2014)</i>	Modified ORCA approach that starts execution with a MILP and it reconfigures to potential fields for all the products (homogeneous) following the perturbation.
<i>Control system CS3 (Pach et al. 2014)</i>	Modified ORCA approach that starts execution with a MILP and it reconfigures to potential fields for the directed affected products (heterogeneous) following the perturbation.
<i>Control system 4 CS4 (predictive-reactive approach)</i>	S-CA approach that follows a predictive technique (MILP) for releasing the jobs and potential fields for machine sequence and routing path decisions.
<i>Control system 5 CS5 (fully heterarchical approach)</i>	Fully static and heterarchical reactive approach for production control LPT-longest processing time for releasing of jobs and potential fields for machine sequence and routing path decisions.

To evaluate the proposed approach in more detail, two analysis phases were conducted. Phase A1 analyses the simulation results of the operating modes in the proposed model (CS1). Phase A2 then compares the makespan of the five different architectures (CS1, CS2, CS3, CS4 and CS5) and presents a detailed comparative analysis of this particular case between our approach (CS1) and the modified ORCA approach with heterogeneous reconfiguration (CS3).

## Results

The results of the experiments are presented in this section. Figure 42 presents the makespan obtained from the simulation for each operating mode for the different scenarios (Phase A1).



**Figure 42** Results of the simulation of different operating modes in the proposed model

Firstly, the results reinforce that performance varies depending on the operating mode used in the control architecture. This variability confirms that the control architecture has reactive behaviour but lacks any real optimality, as it does not choose from a set of alternatives. A RCA that allows different alternatives to be chosen enhances the possibility of absorbing the degradations caused by disruptions. Secondly, these results show that the consequences on overall performance are less detrimental if the perturbation occurs closer to the end of production, as the different operating modes converge to the same performance. We believe that this comes about because when 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, Pollux may calculate the performance indicator of the new 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 decision criterion for the reconfiguration. The results demonstrate the contribution along these lines.

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 no switching	CS5 Fully Reactive no switching
A0	343	370	S1		22	453	455	445	451	454
			S2	DisR-1	195	344	344	344	414	536
			S3		239	343	343	343	385	400
			S4		22	438	438	445*	428	468
			S5	DisR-2	195	344	358	343*	389	400
			S6		239	343	343	343*	370	400
B0	385	401	S7		22	436	451	451	451	454
			S8**	DisR-1	195	389	407	418	442	448
			S9		239	439	458	458	442	445
			S10		22	448	448	429*	507	435
			S11	DisR-2	195	445	485	445*	425	452
			S12		239	405	445	405*	412	452
C0	418	492	S13		22	460	520	484	466	470
			S14	DisR-1	195	462	462	462	529	452
			S15		239	418	486	486	529	452
			S16		22	497	498	559*	529	498
			S17	DisR-2	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

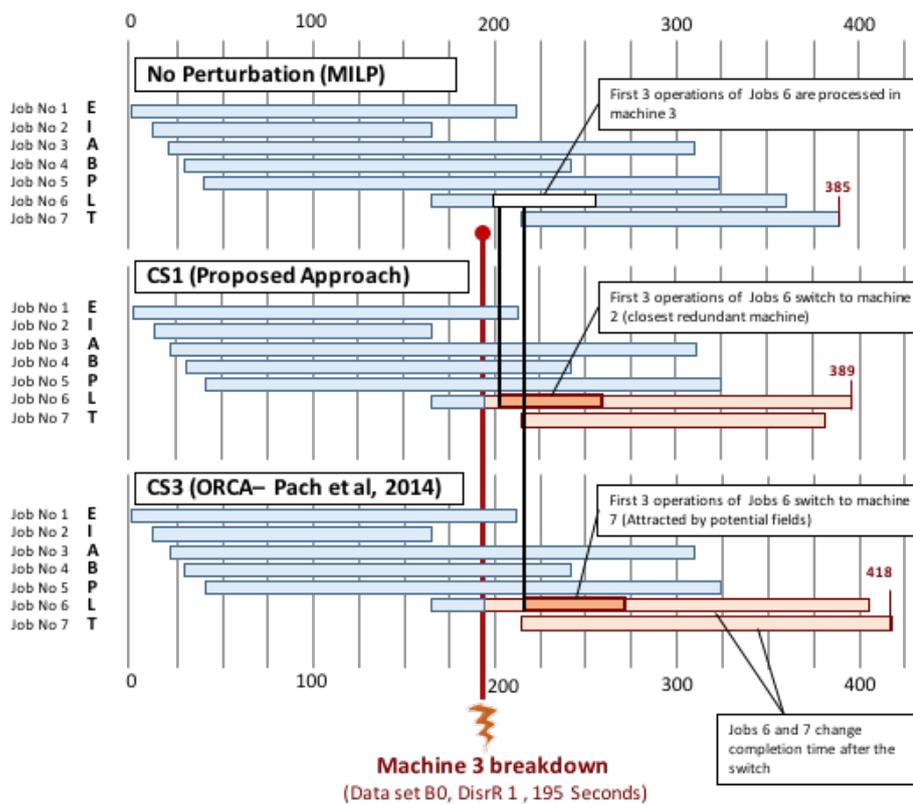


Figure 43 a) Makespan comparison for different control systems (b) Results of the execution of scenario S8

Figure 43 shows a comparison of the makespan of the different control systems, which is part of phase A2. The results obtained show that, under the makespan criteria, the proposed approach (CS1) performs best in six scenarios and shares the best performance in six scenarios. The CS2 approach is not the single best in any scenario but 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 two and one of the 18 scenarios, respectively, while the proposed CS1 behaves better or equals the best for all the others. In this end, the results are explained 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 M3 to M7. The main problem with CS3 is that it uses different resources that were allocated to other jobs, and thus the disruption indirectly affects the rest of the jobs. On the contrary, Pollux considers an operating mode that accounts for 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 (Pollux) compared with 108.57% for the CS3 (ORCA) approach. In this case, the good results are justified, as the reconfiguration technique module explores six different alternatives. On the basis of these simulations, it is worth mentioning that Pollux is beneficial for the reactivity needed in manufacturing systems. From the experiments performed, our approach benefits from the different alternatives derived from the operating modes to adjust its architecture to an optimal configuration. In the other architectures, the reconfiguration is not optimal and reacts only to ensure the continuity of the execution. In addition, it is inferred that the optimal configuration is not a result just from the reconfiguration technique module. In fact, to achieve the optimal configuration, it is necessary to orchestrate the time of change (reconfiguration triggering), the optimisation mechanism (reconfiguration technique) and the implementation of the new control solution (reconfiguration synchronisation).

In experiment A, Pollux demonstrates certain limitations related to the reconfiguration modules. For instance, in the reconfiguration triggering module, it is possible that the mechanism might frequently trigger the change without leaving a stabilisation period, leading to nervous behaviour as defined by (Barbosa et al. 2012). To alleviate this, we suggest exploring a dampening mechanism to address the frequency of reconfiguration. In the reconfiguration technique, with a module depending on the number of alternatives and the complexity of the technique, a large amount of time for reconfiguration might cause further degradation. It is thus worth analysing the balance between the search for the optimal configuration and the time to implement because it may be enough to adopt a merely satisfactory solution instead.

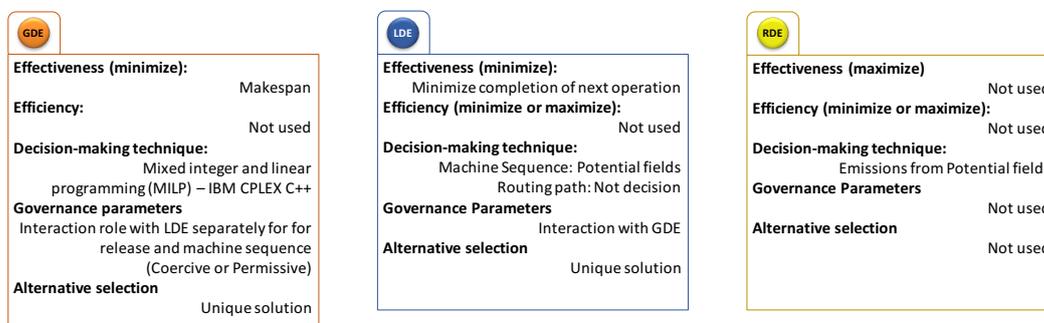
### **5.4.2 Experiment B.**

Experiment B evaluates the application of Pollux on a real full-size academic flexible job shop system. In this study, the global control problem is to set the release sequence, machine sequence and routing path of jobs, and to control the operation during execution. Further information about this experiment is found in Jimenez et al. (2017a).

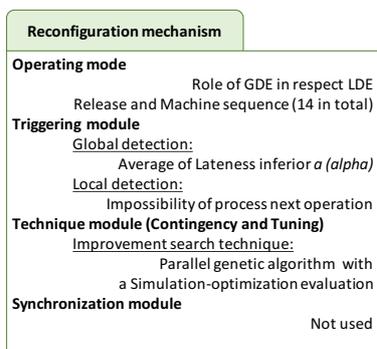
## Conclusions and Future work

**Pollux application:** Pollux is implemented following the general description presented in chapter 3. Figure 44 presents the reconfigurable control architecture (e.g. control architecture and reconfiguration mechanism) and the definition of the decisional entities.

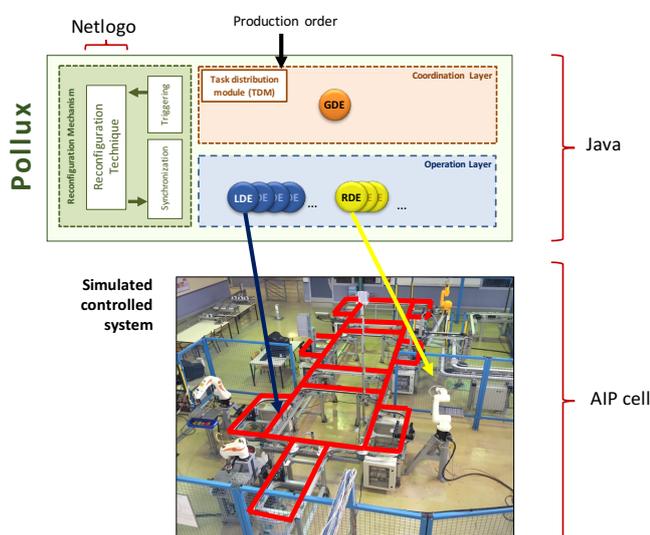
### a) Decision entities for Experiment B



### b) Reconfiguration mechanism for Experiment B



### c) Architecture for Experiment B



\* Each entity represents an object in the AIP cell. This figure do nor relation all objects

Figure 44 Implementation of Pollux for experiment B

## Control architecture

- Coordination layer:** A GDE is created to hold the predictive schedule generation technique (offline) of the jobs to be processed. To do so, it solves the release and machine sequence of each job according to the MILP model formulated in Trentesaux et al. (2013). This was done using the IBM ILog Cplex optimisation studio (IBM ILOG CPLEX 2016) with concert technology (C++). The GDE contains two governance parameters for each LDE to define the pairwise interaction. Whilst the first governance parameter defines the role of the GDE regarding the dispatching of the job, the second governance parameter defines the machine sequence of the LDE.
- Operation layer:** This layer contains  $n$  LDEs as jobs to be produced, and five RDEs as available machines in the flexible job shop system. The LDE decision-making technique is guided by the potential fields approach. The potential fields approach is a reactive control policy technique that guides the routing of the jobs depending on the emission of the potential fields of the RDEs (e.g.

## Conclusions and Future work

machines). This field, which can be attracting or repelling, is dynamically calculated by the availability of the machines (RDEs) that process the requested operation and distance between machine and job (LDEs). The governance parameters of LDEs are an extrapolation from the GDE's governance parameters. For instance, if there are four LDEs, the GDE will have eight governance parameters. In this case study, the values of the governance parameters are coercive (C), to impose GDE intentions on the LDEs, or permissive (P), to provide LDEs with local autonomy.

- **Operating modes:** The operating mode gathers the governance parameters of the GDE, RDE and LDE in order to characterize the configuration of the architecture. The chromosomes for the genetic algorithm in the reconfiguration mechanism is used as the representation of the control architecture. Then, the operating mode is the interactions of the GDE with the LDEs.

**Table 8** Detailed description of decisional entities responsibility and interactions

	GDE	LDE	RDE
<i>Number of Entities</i>	1	$n$	5 ( $M_1, M_2, M_3, M_4, M_5$ )
<i>Responsibility</i>	Predictive scheduling of production order	Guide job execution and coordinate reactive schedules if necessary	Process the jobs and inform status of completion
<i>Decisional components</i>			
<i>Objective</i>	Minimizing production order makespan	Minimize completion time of next operation	Null (not used in case study)
<i>Decision-making technique</i>	Mixed integer linear programming (MILP) programmed in IBM Ilog Cplex	Rational decision: evaluation of available alternatives and proximity to machine for next operation with a heterarchical relation (or following imposed decision with a hierarchical relation)	Null (not decision features)
<i>Governance Parameters</i>	Two for LDE with role in dispatching and machine sequence decisions, Coercive or Permissive (14 in total)	No flexibility in behaviour	Null (not used in case study)
<i>Communication Components</i>			
<i>GDE with...</i>	Not necessary (unique entity)	Coercive imposes MILP intentions. Permissive allows LDE autonomy	Not used
<i>LDEs with...</i>	Information regarding current shop-floor status	Not used	Request the potential field when it reaches a decisional point.
<i>RDEs with...</i>	Report availability	Broadcasts machine availability using potential fields	Not used

### Reconfiguration mechanism

In this case study, the reconfiguration mechanism executes an offline process to set the initial population of the genetic algorithm. The process starts when the data is retrieved from the set of jobs to be processed. These are explained in the reconfiguration technique module.

- **Reconfiguration triggering module:** Because this dissertation does not focus on the triggering module, experiment B implements the same module as experiment A.
- **Reconfiguration technique module:** As stated above, this module has two separate protocols for use at different times during the control execution, called GA-tuning (executed offline) and GA-contingency protocol (e.g. executed online). The GA-tuning protocol is used to form a population of operating modes of the control architecture. It aims to obtain alternative operating modes in

terms of population efficiency and diversity. The GA-contingency protocol periodically improves this initial population based on its fitness function and the current system state. The GA-contingency protocol runs repeatedly and in parallel with execution every  $\Delta t$  depending on the current manufacturing conditions. When a reconfiguration is requested, the GA-contingency protocol changes the operating mode according to the most suitable chromosome from the most recent population. Finally, the system adopts the alignments of the new operating mode to react to the new environmental conditions.

The settings of the GA-tuning and the GA-contingency protocols are presented below. Figure 45 illustrates the use of the genetic algorithm in Pollux's reconfiguration mechanism.

- ✓ **Chromosomes:** the chromosomes for the GA represent a control strategy in the control system. For this, both GA techniques use the operating modes of the control system architecture.
- ✓ **Coding:** the coding in these techniques is used to represent the governance parameters in each decisional entity. Both techniques use a binary value at each gene according to the role of the GDE in the production environment (1 = Coercive role/0 = Permissive role). For example, if the gene of the chromosome is 1 for the first position, the interaction of the GDE with the corresponding LDE (previously allocated) is coercive and the LDE will follow the commands given by the GDE. Conversely, if the gene is 0, the corresponding LDE will act by its own decision-making technique and will ignore the GDE commands.
- ✓ **Population:** the population is a subset of feasible operating modes in the control system architecture. For the GA-tuning technique, twenty chromosomes were selected for the initial population. Since the GA is only used for setting the control system architecture (and not for job scheduling, for example), a previous empirical study for this case study showed that this number of chromosomes was sufficient for our objective. For the GA-contingency technique, the initial population was obtained from the last iteration population of the GA-tuning.
- ✓ **Evaluation chromosome:** this is a simulation model of the same flexible job shop programmed in Netlogo agent-based software (Wilensky 1999).
- ✓ **Offspring selection:** the GA-tuning technique used the restricted tournament method. The GA-contingency technique used the tournament method.
- ✓ **Offspring crossover:** both techniques have a uniform crossover with 50%/50% from parent genes.
- ✓ **Offspring mutation:** both techniques use the bit inversion mutation with a probability  $p_m = 0.3$ .
- ✓ **Stop criteria:** the stopping criteria of the GA-tuning technique were three repetitions of the population average after achieving an optimality gap of 10%. The GA-contingency technique had a limited execution time of  $\Delta t$ .

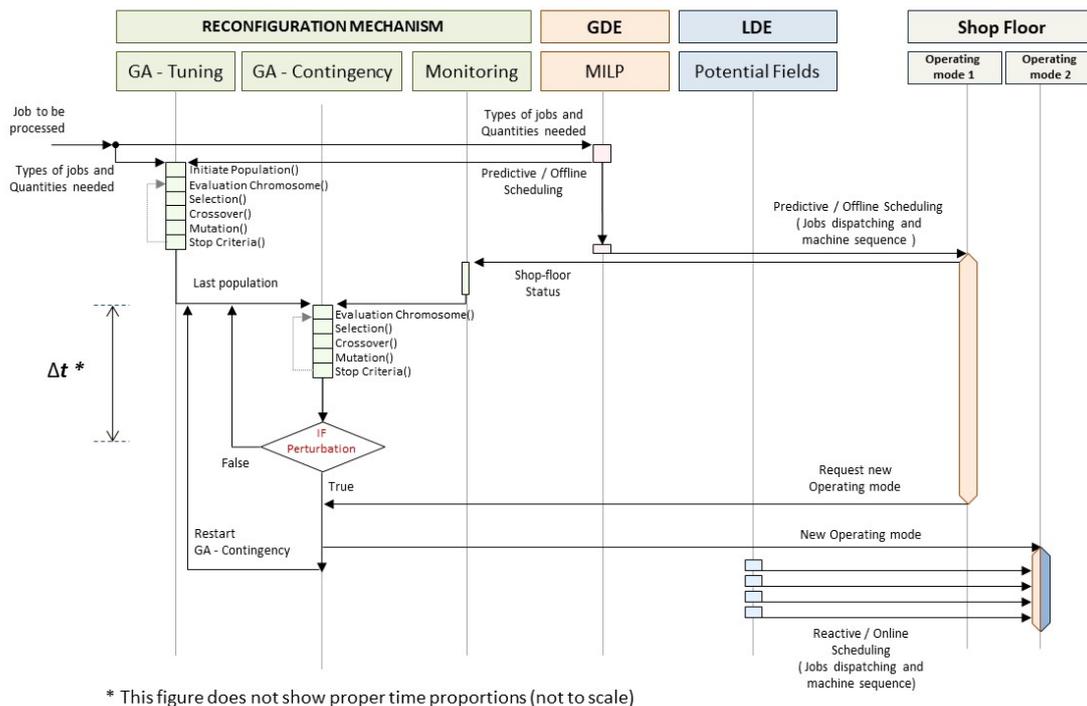
The parameters for the GA-tuning and the GA-contingency is summarised in table 8.

# Conclusions and Future work

**Table 9** Parameters for the genetic algorithms

	GA-Tuning	GA-Contingency
Chromosomes	Defined as the operating modes, as each represents a different control strategy in the HCA. Chromosome of 14 genes where each two genes represent the interaction of the GDE and the LDE. First gene = Role of GDE in dispatching / Second gene = Role of GDE in machine sequence.	
Coding	Binary coding of 14 chromosomes. 1= Coercive role / 0 = Permissive role	
Population	Subset of feasible operating modes in the HCA.	
Initial Population	Selected randomly to allow the entire range of solution be represented in the initial population	Obtained by the last iteration population from the offline procedure (GA-Tuning).
Fitness Evaluation	Evaluated by a Simulation model of the manufacturing cell programmed in NetLogo agent-programming software.	
Offspring Selection	Selection of restricted tournament method	Selection of tournament method
Offspring Crossover	Crossover of uniform crossover (50% / 50% genes of parents)	
Offspring Mutation	Mutation of bit inversion	
Stop Criteria	The is a optimality gap with the MILP solution of less than 5% for 10 consecutive iterations	Execution time of $\Delta t$

- Reconfiguration synchronization module:** Experiment B has an immediate implementation of the operating mode. Therefore, there is no need to synchronise the operating mode or system when it is applied.



**Figure 45** Sequence diagram of the reconfiguration mechanism for experiment B

## Experimental protocol

**Experiment purpose:** the main objective of experiment B is to test Pollux in a real flexible job shop, emphasizing that the inclusion an optimisation algorithm in the reconfiguration process of Pollux improves the efficiency, effectiveness and reactivity in reconfigurable control systems. The case study presented in this section is carried out to evaluate the proposed approach in a real scenario. The

experiments were conducted in the real manufacturing cell located in Valenciennes (France), taking into consideration the benchmark described in Trentesaux et al. (2013). The experiment created four scenarios to test in the AIP cell: scenarios A and B, which are two different instances of applied Pollux with different parameters, and scenarios C and O, which are reference scenarios for comparison purposes.

**Experiment method:** The test used the benchmark for the AIP cell. The data set C0 was tested in the AIP-PRIMECA. The jobs to process are a job of each type (B, E, L, T, A, I and P), and the simulation is limited to four shuttles at the same time. The perturbation considered is a breakdown of machine M3 at 100 seconds after the execution starts. At this point, the machine is disconnected from the system. The reconfiguration process is illustrated in Figure 46. To outline the advantages of Pollux in the experiments, four scenarios were created. In scenarios A and B, Pollux was tested with different  $\Delta t$  parameters for the GA-contingency technique (reconfiguration mechanism). Scenario C was considered as a reference scenario considered in the ORCA approach (Pach et al. 2014). Scenario C starts the production execution with a predictive scheduling technique (the MILP model) and changes to reactive technique (potential fields) at the reconfiguration, heterogeneously. Whilst this change is performed only for the affected jobs, the unaffected jobs remain with the predictive solution. Finally, scenario O is a fully hierarchical architecture with no perturbations.

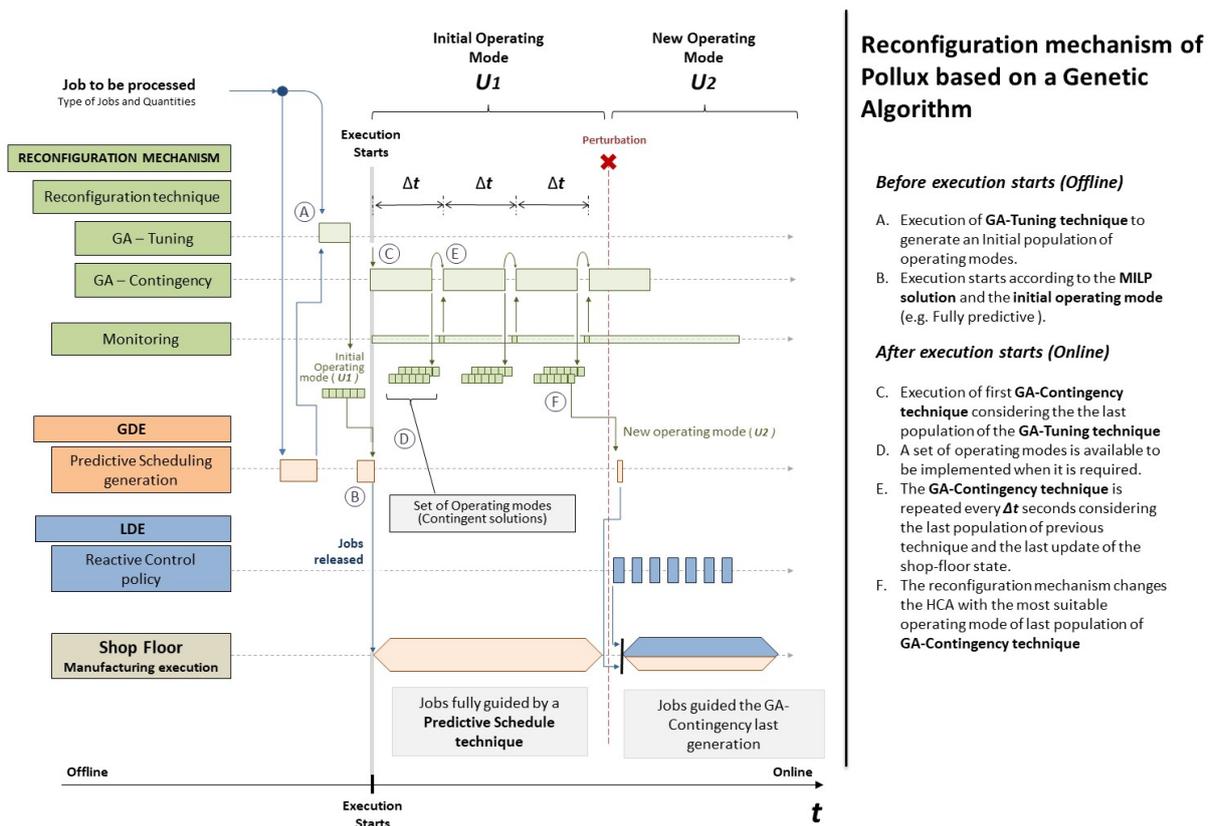


Figure 46 Description of the functioning of the reconfiguration mechanism in the production execution

### Technical implementation of the case study

Consistent with the architecture of Pollux, experiment B was implemented in three layers: the coordination and operation layers from Pollux, and the physical layer, i.e. the real manufacturing system (controlled system). The hardware from the cell was included in level 0 as it holds the physical

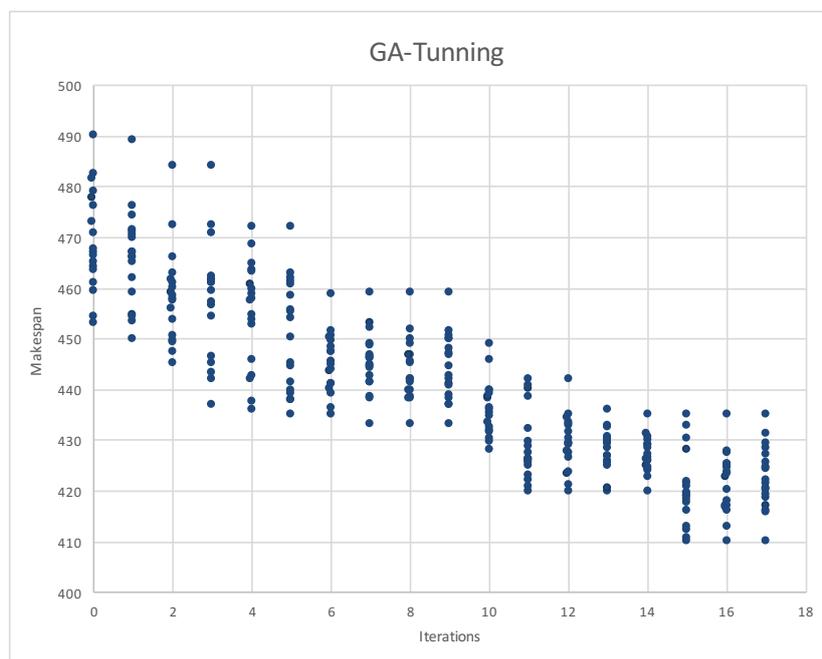
## Conclusions and Future work

layer of the flexible job shop (robots, inspection units, shuttles, transfer gates and positioning units). Level 1 holds the operation layer with a set of networked laptop computers (Asus Eee PC Intel atom1015PEM CPU@ 1.50 GHz with 1.00 GB of RAM memory) as jobs to be processed. Level 2 holds the coordination layer and the GDE in a desktop computer (Intel Core i5-3317U CPU@ 1.70 GHz with 4.00 GB of RAM memory). While each laptop computer runs a Java program with each LDE decisional entity, the desktop computer runs a Java scheduling program for the GDE. The agent-based software Netlogo, located in the desktop computer, is used to simulate the prior evaluation of the fitness of each GA chromosome. IBM Cplex Optimisation software is used to program the MILP model (Trentesaux et al. 2013), where it schedules offline the production order. The three levels are connected via an Ethernet network and they communicate using the TCP/IP protocol. Finally, a WLAN connects the laptop computers and is also connected to the Ethernet network through a router.

### Experimental development

In the first part of the experiment, considering that the only difference between scenarios A and B is the parameter of the GA-contingency technique, it is conducted the GA-tuning technique for these scenarios. The

In the first part of the experiment, considering that the only difference between scenarios A and B is the parameter of the GA-contingency technique, the GA-tuning technique is conducted for these scenarios.



**Figure 47** Results of the GA-Tuning

Figure 47 shows the evolution of the chromosomes (e.g. operating modes) considered in the GA-tuning technique. This technique tunes the initial population of the GA-contingency technique in terms of optimality (average makespan of 423.6 time units) and diversity (standard deviation of 12.51 time units due to restricted tournament selection). The average time to execute each chromosome in Netlogo was 0.287 seconds. The convergence of the GA-tuning technique in these experiments was

## Conclusions and Future work

achieved in 35.72 s (15 iterations). We used these results to set the  $\Delta t$  parameter for the GA-contingency protocol. Therefore, two different instances are defined, as 30 seconds for scenario A and 40 seconds for scenario B, representing two cases before and after the convergence of the GA-tuning algorithm. This definition assumes that the GA-contingency technique behaves similarly to the GA-tuning technique, as it has the same population and its fitness function is evaluated using the same simulation tool (Netlogo).

In the second part of this validation, the four scenarios were tested in the real flexible job shop. In scenario O, the jobs resulted in a makespan of 404 s. In scenario A, the production of the jobs was conducted implementing Pollux with a GA-contingency technique every 30 s. To this end, the reconfiguration mechanism retrieved the data from the flexible job shop every 30 s to readjust the initial state of the Netlogo Model. After the perturbation, the makespan of scenario A was 457 s (13.12% degradation). In scenario B, where the GA-contingency technique was executed every 40 s, the makespan was 466 s (15.34% degradation). In scenario C, the makespan was 504 s (24.75% degradation). Figure 48 illustrates the results of the experiments and compares the results with an existing approach (ORCA).

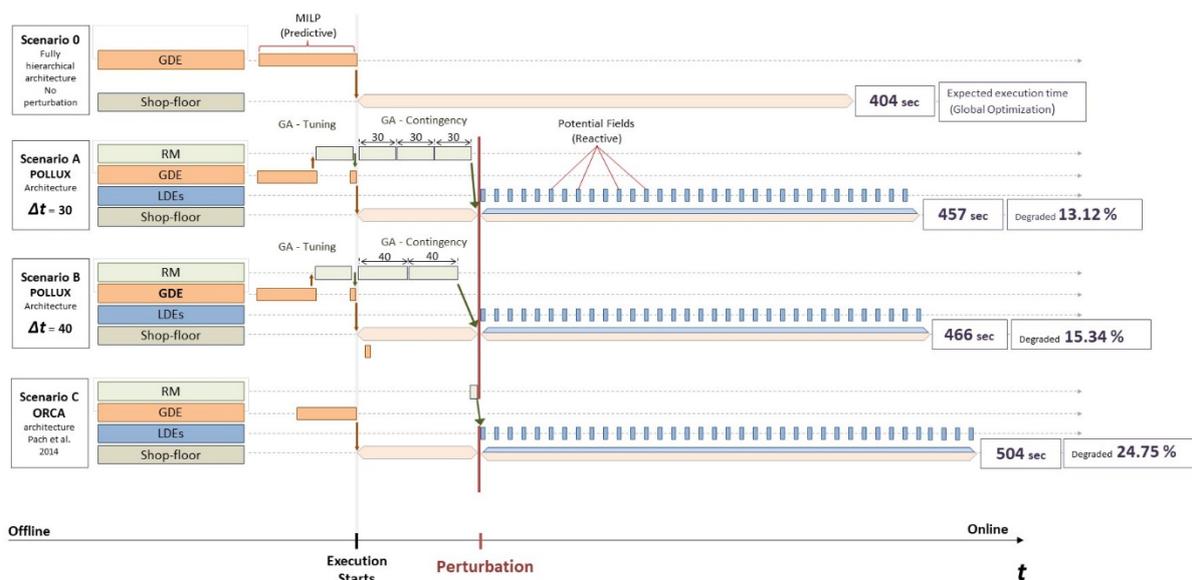


Figure 48 Makespan comparison for three different execution strategies

**Experiment Results:** The results demonstrate that Pollux has a better performance than the reference approach (ORCA). In this scenario (scenario C), although it is observed that jobs unrelated to M3 (the perturbed resource) are not directly affected, the perturbations have an indirect impact, as affected jobs employ the previously available resources in the AIP cell. Consequently, no affected jobs start looping through the cell searching for the same predictive intention. On the contrary, in scenarios A and B, it was observed that the new operating modes adjusted better to the perturbed conditions, as the new operating mode pre-evaluates the performance and considers the behaviour of affected and unaffected jobs simultaneously.

From experiment B, three conclusions may be drawn. First, an improvement in the reactivity requirement was demonstrated in situations where Pollux adjusted the control solution to a better operating mode. To this end, Pollux demonstrates that reactivity can be achieved at both low and high levels, through the orchestration of decisions to both affected and unaffected jobs. Second, the inclusion of a reconfiguration mechanism with an evolutionary technique (the optimisation technique) represents a promising area of research for reconfiguration control systems. In this experimental case study, this parallel reconfiguration mechanism provides a set of alternatives to apply in response to unexpected events, avoiding reconfiguration delays. It is recognized that this technique requires further research, as many problems can occur, including the inability to find a suitable operating mode from the set of alternatives due to the nature of a perturbation. Nonetheless, in this case study, the reconfiguration mechanism searched a diverse population to find repair alternatives to apply under a perturbation. Finally, this research helps prove that Pollux features a dynamic shared autonomy between the hierarchical (coordination layer) and heterarchical (operation layer) control. To this end, the inclusion of a reconfiguration mechanism is appropriate, as this shared autonomy must be adjusted throughout the production execution in order to achieve the improvement of efficiency and effectiveness needed after a perturbation.

### 5.4.3 Experiment C.

Energy is one of the most significant metrics to address in sustainable manufacturing, as industries consume a significant portion in the global economy (Prabhu et al. 2015). Industries have traditionally disregarded the optimisation of energy consumption as it was considered as a low-cost and unlimited resource that had a negligible impact on manufacturing operations. However, eco-friendly concerns, inflated energy costs and market price dynamics have now increased the interest in integrating energy awareness into manufacturing decision-making (Giret et al. 2015). An example of this awareness and its consequences in manufacturing decision-making is the price-based demand in energy contracts. Such contracts, currently in place in several European countries, have short-term price shifts in energy for influencing power consumption (Detzler et al. 2015). Consequently, industries optimise consumption by adapting their operations according to these fluctuations. This case study focusses on this area of application, as manufacturing operations can currently be disrupted by such energy-price fluctuations.

In experiment C, Pollux is applied in a sustainable application where it is integrated with sustainability objectives (e.g. environmental, social and economic) of the reconfigurable control system. To this end, Pollux involves multi-criteria objectives from the sustainability perspective in both the control architecture and the reconfigurable mechanism. Therefore, objectives may be set specifically for the efficiency or effectiveness of the manufacturing operation. For example, effectiveness goals may include the optimisation of the makespan, throughput or order tardiness, while efficiency goals may be production and inventory costs in economic terms, labour practices and community involvement for the social dimension, or total energy consumption and CO<sub>2</sub> emissions for the environmental dimension. This experimental case study focusses on the environmental dimension, and specifically energy consumption.

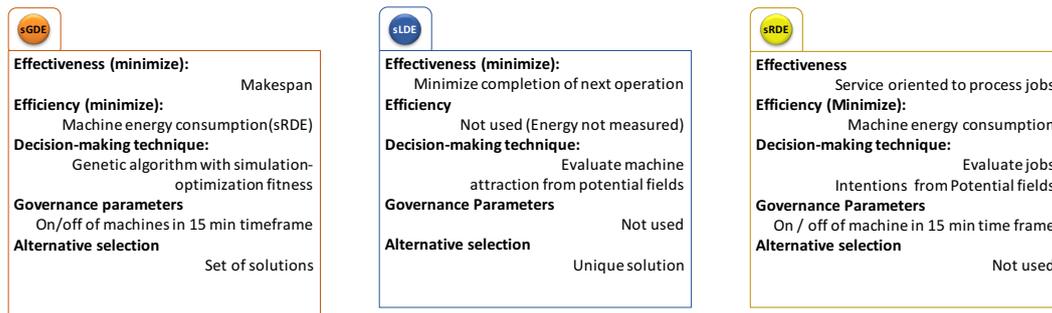
**Pollux application:** For Experiment C, Pollux is implemented following the general instantiation described above chapter section 3. Figure 49 presents the reconfigurable control

## Conclusions and Future work

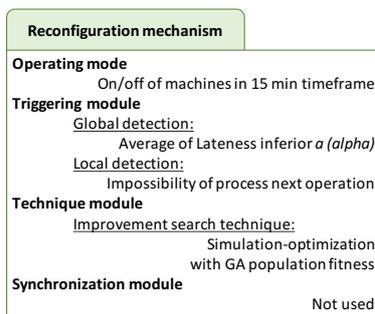
architecture (control architecture and reconfiguration mechanism) and defines the decisional entities. In this experimental case study, the decisional entities have a multi-criteria objective, in terms of efficiency and effectiveness, in order to model an energy-aware case study of the proposed approach. A further analysis of this case study is given in Jimenez, Bekrar, Giret, Leitao & Trentesaux (2016b).

The global control problem of this Pollux application, called Sustainable Pollux or sPollux, is to manage the regular decisions from the FJSP (e.g. release sequence, machine sequence and routing path of jobs) and the machines turning on/off at intervals of 15 minutes (e.g. only the redundant machines).

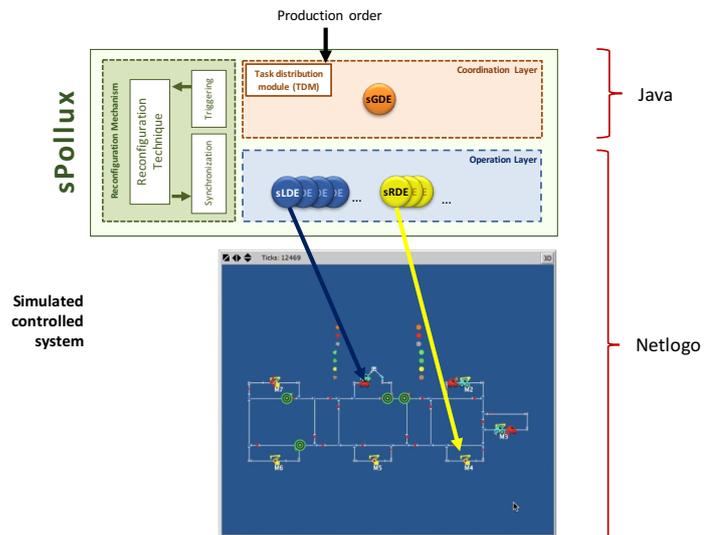
### a) Decision entities for Experiment C



### b) Reconfiguration mechanism for Experiment C



### c) Architecture for Experiment C



\* Each entity represents an object in the AIP cell. This figure do nor relation all objects

Figure 49 Implementation of Pollux for experiment C

### Control architecture:

- Decisional entity:** the decisional entity of sPollux is modelled as a Go-green manufacturing holon. Go-green manufacturing holons (see figure 50) are autonomous and cooperative holons that represent physical manufacturing entities (products, machines, conveyors, AGVs, etc.), whose

decisions are balanced through a trade-off between efficiency and effectiveness indicators required to undertake activities in sustainable manufacturing objectives (Trentesaux & Giret 2015).

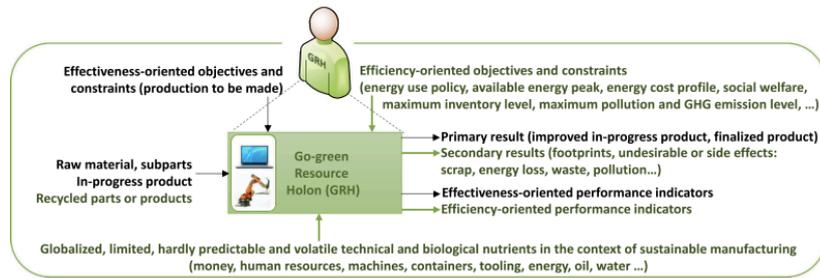


Figure 50 Go-green holon in manufacturing (Trentesaux and Giret 2015)

- Coordination layer:** the coordination layer hosts a GDE. This GDE is called sustainable-GDE or sGDE, as it has both efficiency and effectiveness objectives. The sGDE provides the release sequence, machine sequence and routing path of the jobs, and sets the energy management during the operation. The sGDE hosts a multi-objective genetic algorithm (called MO-GA) whose fitness function is a simulation model developed in the Netlogo agent-based simulation software (Wilensky 1999). In this multi-objective problem, the fitness function is an aggregated weighted trade-off between the makespan ( $s$ ) and the energy consumption ( $W-h$ ) calculated with a weighed sum method. This is done using weights  $W_{C_{max}}$  of 0.226 and  $W_{Energy}$  of 0.774, assigned empirically in order to normalize the values from the effectiveness and efficiency objectives.
- Operation layer:** the operation layer hosts the sustainable local decisional entities (sustainable-LDE or sLDE) and sustainable resource decisional entities (sustainable-RDE or sRDE), which are also modelled as Go-green manufacturing holons. Each sLDE, which represents the jobs associated with each production order, is responsible for managing the job behaviour based on a reactive/online control approach. Each sRDE which represents the machines handles the processing activities based on a service-oriented approach. While the sLDE resolves the machine sequence and routing path of jobs through attractive potential fields, the sRDE resolves the machine turn on/off decision through an intentional potential field. The potential fields technique (Zbib et al. 2010) is a reactive technique where a decision is made according to fields emitted by a corresponding entity. For attracting potential fields, the sLDE evaluates the attraction-value acknowledged by the machines (e.g. availability and proximity) in order to be directed reactively to the next machine. For the intentional potential fields, the sRDE evaluates the intentions acknowledged by jobs (e.g. frequency and duration) in order to reactively turn on/off the corresponding machine online.
- Operating mode:** The benefit of using a genetic algorithm is that it is a population-based metaheuristic that constructs several offline schedules with similar fitness values but with different effectiveness and efficiency objectives. Consequently, each individual from the last iteration of the GA is used to represent a possible operating mode for production execution. While the best individual is used as the initial operating mode and starts the production execution, the other individuals are kept as alternative operating modes to be used in case of perturbations.

### Reconfiguration mechanism

- **Reconfiguration triggering module:** Because this dissertation does not focus on the triggering module, experiment C implements a similar module to that of experiment A. However, the threshold that activates the reconfiguration is related to both the effectiveness and efficiency metrics. Contrary to the previous experiments, Pollux monitors the effectiveness with throughput in a defined timeframe, and efficiency by energy consumption measured in that timeframe. Here the effectiveness and efficiency thresholds are defined as 15% of the throughput and the energy consumption (each 15 min) in the predictive-offline solution without perturbation. The throughput is chosen as the classic local performance indicator during the execution. For example, if the throughput per timeframe is below a certain threshold, Pollux will change the operating mode to favour the effectiveness indicator at the expense of the efficiency one. Likewise, if the energy consumption average is exceeded, Pollux will change the operating mode to favour the efficiency regardless of the effectiveness.
- **Reconfiguration technique module:** This module is implemented because the operating modes are fixed by the sGDE in the offline phase. Once the MO-GA is executed, the resulting population is considered a set of operating modes to apply when a reconfiguration is requested. The reconfiguration technique module must only evaluate this set of operating modes to choose the best alternative. Certainly, to consider this situation, this case study assumes that the only perturbation is the change in the price rather than a breakdown or any other event. Therefore, if the thresholds of the triggering module are crossed, the reconfiguration technique module chooses an operating mode by an improvement-search heuristic based on a trade-off between efficiency and effectiveness indicators. This is done through a simulation-based optimisation using a modified Netlogo model where the throughput and the energy consumption are measured.
- **Reconfiguration synchronization module:** Experiment C has an immediate implementation of the operating mode. Therefore, there is no need to synchronise the operating mode or system when it is applied.

### Experimental protocol

**Experiment purpose:** The main objective of experiment C is to analyse the efficiency, effectiveness and reactivity of sPollux subject to the price shift perturbation. It encompasses evaluation of the controllability and optimality achieved in scenarios with multiple objectives, and tests a sustainable version of Pollux in an energy-aware approach. The simulations are carried out in a Netlogo model with energy metrics, which measures the throughput and energy consumption of the simulated AIP cell.

**Experiment method:** the case study included 490 jobs (70 units of each letter, A, I, P, B, E, L, and T), which are guided by shuttles within the conveyor of the FMS. This production order of 490 jobs was chosen due to the empirical production time without disruptions of between 5 and 7 hours. The usual global control problem to address in this FMS is to resolve the release sequence, machine sequence and routing path of jobs in the AIP cell. In order to integrate sustainability issues in the proposed case study, an additional decision, turning the machine on/off, is included in this problem since this brings about a significant reduction of energy consumption in manufacturing operations (Mouzon et al. 2007). Therefore, a machine may be off, idle or working during execution. However, to reduce the complexity and bound the problem flexibility in this case study, this new decision is considered only for redundant

## Conclusions and Future work

machines (M4 and M7).

In addition, and to be consistent with the introduced context dealing with the future of energy management in industries, the experiment considers an energy contract as a *price-based demand*. Although a forecasted price can be estimated during the workday, a shift of the energy price is considered as a perturbation and is limited to 10% of the predictive price. The perturbation time is simulated stochastically by a uniform function  $U(0, \text{predictive } C_{max})$ .  $C_{max}$ , also known as the makespan, is the completion time of the last job processed in the production order. Figure 51 illustrates the functioning of sPollux applied to a FMS.

The experiment included 10 different cases with different perturbation times. The experimental protocol was then carried out in three different control solutions: approach A, a predictive model without perturbation, approach B, a predictive-reactive approach without a reconfiguration, and approach C, implementing sPollux. In sum, these control solutions are compared and the specific case study is detailed.

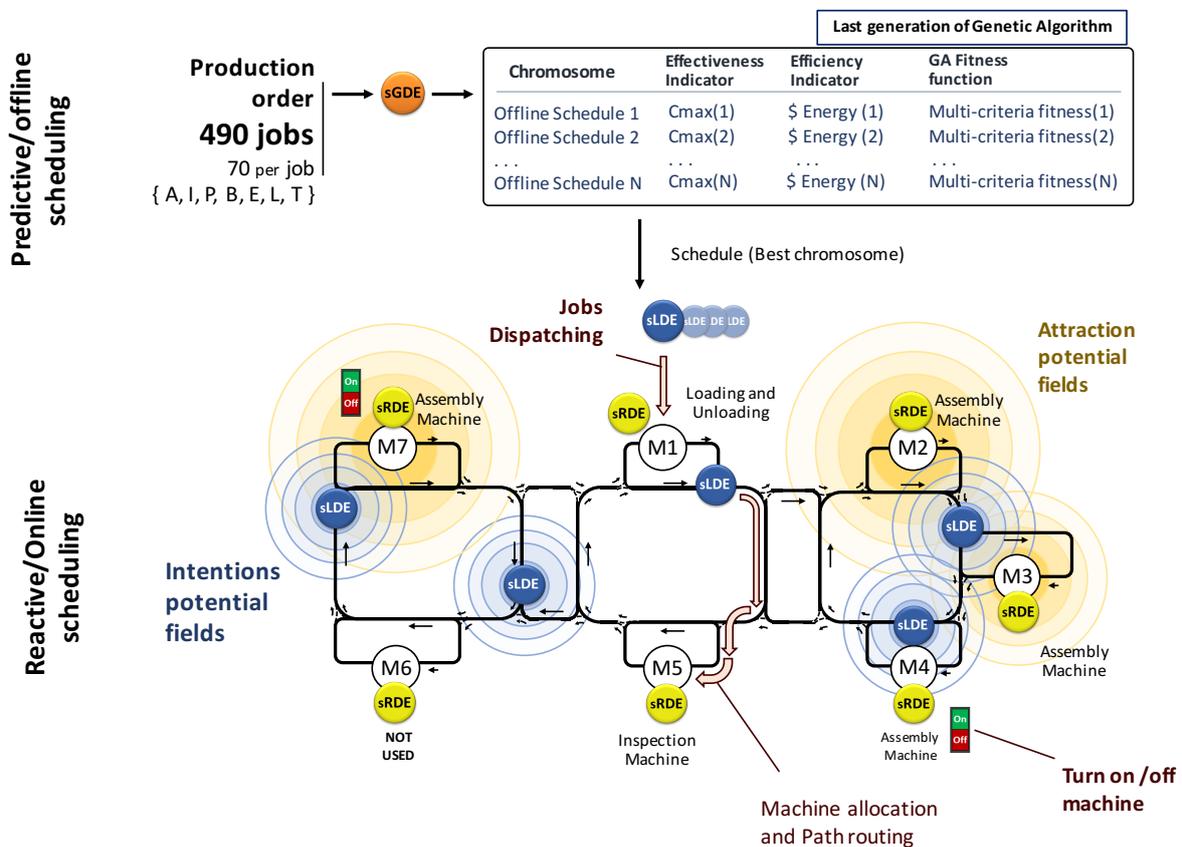


Figure 51 Schematic view of the experimental case

### Experimental results:

Approach A executes the experiments using the same MO-GA from sGDE, and not considering any perturbation. This approach is taken as a reference scenario in order to measure the degradation of the system in comparison with other scenarios. For the simulated execution, the  $C_{max}$  and the energy consumption were 15460 seconds (simulation seconds) and 4528 W-h respectively. In addition, this

## Conclusions and Future work

execution is taken into account in order to tune the thresholds of the reconfiguration mechanism of Pollux. These thresholds are defined according to the average throughput and energy consumption in time frame  $T$  of 15 min. More specifically, the effectiveness and efficiency thresholds are 25.2 jobs/timeframe (29.6 minus 15%) and 315.2 W-h/timeframe (274.1 plus 15%) counting from the fourth timeframe (from 0.75 h to 1 h) as it is reached the metric's convergence. For approach B, the predictive-reactive approach is executed 10 times in a Netlogo model setting each defined perturbation. The architecture of this approach is a static-CA with a predictive coordinator (the same MO-GA used for sGDE) and a potential-fields technique for the reactive guidance of the products. In this static-CA, the control system does not reconfigure the control architecture and it absorbs the perturbation completely with the reactive technique. These experiments were conducted to evaluate if a reconfiguration with effectiveness and efficiency considerations does or does not enhance the responsiveness of a control system. In approach C, sPollux is executed for the 10 scenarios. Table 10 shows the experiment results for approaches A, B and C. In this table, the percentage for Cmax corresponds to the final degradation of the metric after the perturbation compared to the predictive solution and for the energy it corresponds to the percentage of energy consumed compared to the predictive solution (A vs. C). In addition, we may compare the percentage of degradation for the predictive/reactive solution and the proposed solution (B vs C).

**Table 10** Comparison of approaches after the perturbed scenario

Scenario	Performance Indicator	Approach A	Perturbation Time ** (Price Shift)	Approach B		Approach C	
		Predictive Model (No perturbation)		Predictive-Reactive Model *	% ***	Sustainable Pollux Model *	% ***
1	Cmax	15460	1070 s	17184	11.2%	16125	4.3%
	Energy	4621		4458	96.5%	4534	98.1%
2	Cmax	15460	2881 s	17113	10.7%	15621	1.0%
	Energy	4621		4452	96.3%	4490	97.2%
3	Cmax	15460	3975 s	17126	10.8%	15840	2.5%
	Energy	4621		4453	96.4%	4505	97.5%
4	Cmax	<b>15460</b>	<b>6022 s</b>	<b>17029</b>	<b>10.1%</b>	<b>15938</b>	<b>3.1%</b>
	Energy	<b>4621</b>		<b>4442</b>	<b>96.1%</b>	<b>4516</b>	<b>97.7%</b>
5	Cmax	15460	8737 s	15970	3.3%	16906	2.9%
	Energy	4621		4487	97.1%	4612	99.8%
6	Cmax	15460	9854 s	16214	4.9%	15920	3.0%
	Energy	4621		4481	97.0%	4514	97.7%
7	Cmax	15460	10663 s	15780	2.1%	15889	2.8%
	Energy	4621		4528	98.0%	4511	97.6%
8	Cmax	15460	10951s	15724	1.7%	15828	2.4%
	Energy	4621		4592	99.4%	4354	94.2%
9	Cmax	15460	13740 s	15664	1.3%	15851	2.5%
	Energy	4621		4581	99.1%	4507	97.5%
10	Cmax	15460	14747 s	15664	1.3%	15802	2.2%
	Energy	4621		4590	99.3%	4601	99.6%

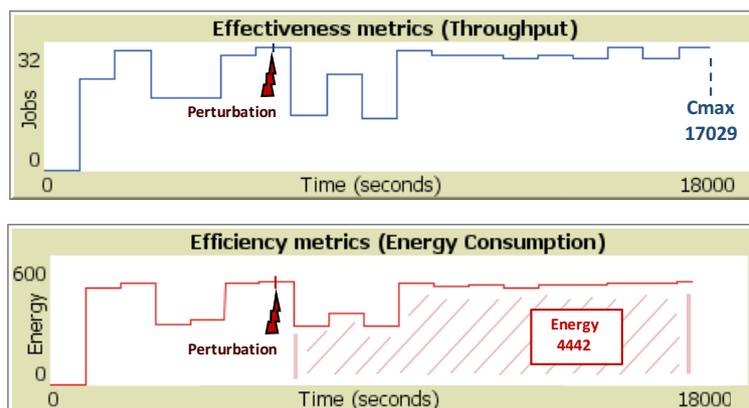
\* Cmax in seconds and Energy in Watts-Hour  
 \*\* Organized upwards  
 \*\*\* Cmax: Degradation from perturbation. Energy: Percentage consumed compared from predictive solution

In addition to these results, the execution of scenario 4 is detailed and compared. First, the execution according to the initial operating mode is launched (e.g. best chromosome of the GA in terms of

## Conclusions and Future work

aggregate fitness). During the execution, the perturbation time for simulation 4 has been randomly set at 6022 seconds. The reconfiguration mechanism detects at 6300 seconds (i.e. at the end of the time window) that the energy consumption indicator has exceeded the energy threshold. At this time, the reconfiguration mechanism triggers the improvement search heuristic to change the operating mode. After the calculations, it finds a new operating mode and performs a synchronization according to the work-in-progress jobs. Finally, the newly found operating mode is applied. Figure 52 illustrates the experiment execution for simulation 4.

### a) Simulation for Scenario 4 (Predictive –Reactive model)



### b) Simulation for Scenario 4 (sPOLLUX model)

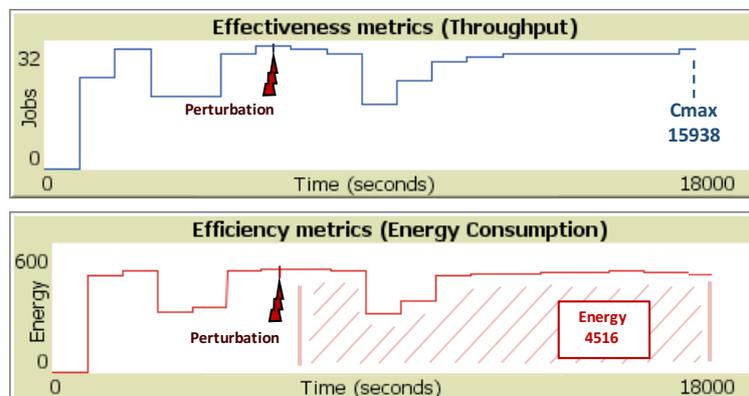


Figure 52 Simulations of scenario 4 for Approach B & C.

From these experiments, it can be seen that sPollux (approach B) provides better performance after a perturbation than the predictive-reactive approach (approach C). We believe that these results are satisfactory because Pollux has three complementary mechanisms that respond to the energy fluctuation. First, it considers an alternative near-optimal individual from a predictive-based trade-off (sGDE). Second, it has a default execution from the reactive-based trade-off when the perturbation is not yet detected (sLDE). Third and finally, it has a reconfiguration mechanism that adapts a new operating mode to the actual needs of the system (reconfiguration technique module).

## 5.5 Summary

This chapter describes the implementation and validation of the proposed approach in two simulated and one real scenario of a FMS located in the University of Valenciennes (France). Initially, the description of the FMS is presented in order to set the case study for the conducting experiments. Then, the general implementation of Pollux is parametrized according to the FJSP. Three experimental scenarios were conducted in this chapter to assess the proposed approach. First, the experimentation showed the feasibility and diversity resulted in having different operating modes in a control architecture. Secondly, it presented the implementation of optimality-based reconfiguration process for a real manufacturing system. Finally, it introduced an application of Pollux for dealing with indicators that measure the sustainability development in the control execution.

The experiments show that the proposed approach performs better respecting to other predictive-reactive approaches. Pollux is capable to achieve the stability and recovery of the control system by assigning different responsibilities to the components of the control architecture. First, the optimality of the operations is achieved by a global performance component that achieve the optimal results through predictive approaches. Secondly, the reactivity of the operations and control architecture is achieved by the distribution of reactive techniques into an agent-based architecture that monitors local functioning. Certainly, the optimality and reactivity of operations are specific objectives of the controlled system. And third, the reconfiguration mechanism is capable of responding optimality to the system perturbations as it can control the entire system through the governance mechanism.

# Conclusions and future work

Reconfigurable control systems are a part of the ICT revolution that offers a clear opportunity to build the efficient, effective and reactive systems required in modern society. These are not only capable of setting the parameters, components, and behaviour of a control architecture before it starts the execution of the system, but are also capable of changing these characteristics of the control architecture automatically and in real time in order to adjust the control and controlled system and minimise the degradation of global performance caused by unexpected events. In addition, while it is an efficient reference architecture that is capable of managing several complex scenarios from different domains, it can be used to leverage and cooperatively integrate new technological advances provided by this revolution (e.g. cyber-physical systems, artificial intelligence, automation, etc.). The need, therefore, is to design and develop reconfigurable control systems that orchestrate the hardware, software and human-interaction aspects in real time during system execution.

The aim of this dissertation is to create a reconfigurable control system that manages and adjusts, optimally and in real time, the architecture of a control system, both for guiding the operational execution and as a response to unexpected changes in a controlled system. The main goal was to build and assess a reconfiguration mechanism that includes optimisation-based principles in the reconfiguration process in order to improve the controllability and optimality of the reconfiguration. To this end, our contribution started by defining a framework for the optimal configuration of the architecture of a semi-heterarchical control system. This framework takes into account that the structural and behavioural characteristics and the combinatorial possible states of these characteristics are an architectural organisation that defines the mode of functioning (or configuration) of a control system. The gain is that each configuration of this semi-heterarchical architecture provides a diverse architecture in between a hierarchical and heterarchical arrangement, as well a particular performance indicator. Therefore, the framework shows that the change of configuration and subsequent performance can be encouraged throughout the changes of the structural and behavioural characteristics of the control architecture. From this starting point, this dissertation set out to make the following contributions.

The first contribution concerned the findings from the literature review. First, it was found that reconfigurable control systems are a promising approach that enhances the capabilities of control systems and provides a tailor-made configuration and solution for any purpose. Second, we found that the implementation of reconfigurable control systems needs a flexible and customizable control architecture that provides combinatorial alternatives for a posterior reconfiguration and orchestration process. Third, it was confirmed that there is a need to optimise reconfiguration to ensure the adoption of the best possible control solution. However, considering that the optimisation technique might need considerable execution time, it was found that the challenge is to provide a reconfiguration technique that strikes the proper balance between reconfiguration optimality and reconfiguration reactivity. Fourth, as a knowledge gap in the field of discrete-event control systems, it is not yet clear how the reconfiguration process can achieve optimality changes after a perturbation. Finally, the literature review revealed the need to answer the following questions: *When*: the best time in the execution to

perform a reconfiguration must be known. **What**: it must be known exactly what should be changed in the control architecture to achieve recovery during the reconfiguration. And **How**: it must be known how the identified component(s) should be changed in order to reach a new and improved configuration in terms of the control solution.

The second contribution was the definition of a model of decisional entities that are flexible and customizable enough to personalize the behaviour and interaction with other entities. These entities contain a specific set of parameters, called the governance parameters that can be changed repeatedly to set the rules of conduct for each decisional entity. In this model, the definition of these governance parameters makes it possible to establish any parameter from the decisional entity (i.e. objective to follow, decision-making technique to run, evaluation criteria, etc.). However, this dissertation focusses on the flexibility provided in the interactions between the different components, setting the governance parameters as the type of relationships (e.g. hierarchical, modified-hierarchical or heterarchical relationship).

The third contribution was the creation of a model for an operating mode that serves as a representation of a specific configuration, characterizes the control solution, permits comparison between different operating modes, and can give estimates of its performance during application. Considering that the optimisation reconfiguration framework is the starting point of this proposal and the governance parameters make it possible to customize the behaviour of each decisional entity, the operating mode is the representation of the architecture that allows us to set the mode of functioning of the control system.

The fourth contribution was the construction of a reconfigurable control architecture that is flexible and customizable enough to personalize with the needed predictive, reactive and reconfiguration techniques. This architecture consists of decisional entities (i.e. global, local and resource decisional entities), which are organized in coordination and operation layers and interact and participate in a collective emergent process to meet global and local objectives. This contribution is the basis for the improvement in the controllability of the system. The flexibility and customization features of the reconfigurable control architecture make it possible to change the parameters and thus make it possible to set the configuration to a desired resulting state.

The fifth contribution was the definition of a three-module reconfiguration mechanism that executes the reconfiguration process when needed. This framework, which responds to the **when**, **what** and **how** questions found in the literature review, supports the reconfiguration process by monitoring the system in the event of a perturbation, triggering the reconfiguration when needed, changing the configuration of the control architecture according to the corresponding system's needs, and implementing the new configuration to continue with system execution. The modules are the reconfiguration triggering, reconfiguration techniques and reconfiguration synchronization.

The sixth and main contribution was the definition of a general procedure for the reconfiguration technique, which considers the optimality-based principles to execute the reconfiguration process in the event of a perturbation. The reconfiguration technique is a specific protocol that includes the representation of alternatives, the evaluation of alternatives, the iterative process, the stop-criterion, and the criteria for choosing from the alternatives. This contribution is the basis for the optimality of the system. In short, the main advantage of this technique is that it contains an iterative improvement search process that strategically searches within different operating modes to

find an optimal configuration. Depending on the degree of optimisation required, this technique may search for an optimal, near-optimal, towards-optimal or heuristic solution for the reconfiguration.

The proposed reconfigurable control system is applied in the manufacturing domain and is validated in a simulation and in the real cell of a flexible manufacturing system located in Valenciennes, France. To this end, three experimental scenarios designed to demonstrate the flexibility and customization features of Pollux were conducted to assess the proposed approach. These three experiments showed the feasibility and diversity brought about by having different operating modes in a control architecture, included an optimality-based reconfiguration process for implementing the best alternative after the reconfiguration. They served as a case study for implementing the reconfiguration mechanism in parallel to the execution, and introduced a version of Pollux for addressing sustainable development in the control execution.

The scope of this study was limited to the reconfiguration technique rather than the monitoring, triggering and implementation of the new configuration. In addition, it intended to set the orchestration of multiple technologies rather than conducting new theoretical research in this area. However, although the improvement in efficiency, effectiveness and reactivity is achieved through the reconfiguration mechanism, it is clear that the time detection of the perturbation and implementation of the new operating mode are factors that might impact the overall performance stability and recovery. In addition, this approach has some limitations related to the reconfiguration modules. In the reconfiguration technique, it is possible that the mechanism might frequently trigger the change without leaving a stabilisation period (leading to nervous behaviour). To alleviate this, it is recommended to explore a dampening mechanism to limit the frequency of reconfiguration. Furthermore, in the reconfiguration technique module, depending on the number of alternatives and the complexity of the technique, an extensive time for reconfiguration might cause further degradation, leaving the synchronization unable to be repaired. To this end, it is worth analysing the balance between the optimal technique and the reconfiguration time to adopt a satisfactory solution.

Several lines of research may be derived from this work. These are related to the framework for the optimal reconfiguration, the reconfiguration mechanism, and general issues of the reconfiguration control system. Within the framework for the optimal reconfiguration, it is clear that theoretically each configuration or operating mode has a degree of governance in terms of the hierarchical-predictive and heterarchical-reactive architecture. However, the exact degree of governance is not defined, and this makes difficult to evaluate the operating mode without an already existing evaluation of the operating mode. Therefore, it would be interesting to create a methodology for measuring the degree of governance of the control architecture between the global and local entities. Although an efficient evaluation technique is found in a simulation-based optimisation, this degree could be an estimation of the shared control and might accelerate the reconfiguration process.

With regard to the reconfiguration mechanism, and more specifically the reconfiguration technique module, there are several research perspectives that would be interesting for improvement of the performance of the reconfiguration control systems. First, further research is needed to better understand the balance between the reconfiguration optimality and reconfiguration execution time. During the reconfiguration process, it is necessary to choose the stop-criteria of the optimisation technique. However, the time of the optimisation technique directly impacts the reconfiguration execution time, and consequently deepens the degradation of the system. To this end, we suggest

## Conclusions and Future work

---

exploring a trade-off mechanism to balance these two reconfiguration parameters. Second, considering that it is semi-heterarchical architectures which contain redundancy processes for reactivity objectives, it would be interesting to include a mechanism that evaluates the cost of reconfiguration per unit of execution time. This mechanism or other system evaluation methods should be carried out to establish the parameters of the reconfiguration mechanism. Third, the reconfiguration triggering module of the reconfiguration mechanism can change frequently without leaving a stabilisation period, leading to nervous behaviour. Further research could therefore usefully explore how to include a dampening mechanism that controls this over-triggering and stabilizes the dynamism of the control system.

As concerns the triggering and the synchronization modules, the dissertation did not assess the impact of these protocols on the overall reconfiguration process. A future study investigating the interrelation among the three reconfiguration modules would be very interesting. For example, the triggering module needs to deploy analytics for monitoring the system. This module may therefore deal not only with data reporting and current/trends analysis, but also with forecasting and predicting the possible behaviour of the system in the future. In this regard, the three modules of the reconfiguration might work in synergy to improve the performance of the reconfiguration process.

With regards the resolution of societal needs by technological advances, this dissertation has raised important questions regarding the complexity factors in the design and implementation of the technological solutions. In particular, there are two factors that will be very interesting due the complexity level. On one hand, the human-in-the-loop is recently increasing the research perspectives. It involves mainly the level of inclusion of the human within a technological solution. On the other hand, the inclusion of ethics in the technological solution. It involves the distribution of responsibilities regarding the technological solution. In recent years, there has been an increasing interest in these two factors, for Human-in-the-loop (Schirmer et al, 2013) and Ethical inclusion (Trentesaux, & Rault, 2017b).

Finally, I suggest that further research should be undertaken to test the proposed approach in another context. This implies implementation in other domains (i.e. logistics, health-care, electronics, agriculture, telecommunications, etc.), inclusion of various multi-criteria objectives (i.e. economic, social and environmental), other types of problems with different decisions and flexibility (i.e. allocation planning, path routing, facility allocation, network optimisation), and other control problems (i.e. project management, airline control, social networks, etc.).

# References

- Alencar, P.S. and de Lucena, C.J., 1996. A logical framework for evolving software systems. *Formal Aspects of Computing*, 8(1), pp.3-46.
- Alur, R., 2015. Principles of cyber-physical systems. MIT Press.
- Aström, K.J., Albertos, P., Blanke, M., Isidori, A., Schaufelberger, W. and Sanz, R. eds., 2001. *Control of complex systems*. Springer Science & Business Media.
- Atzori, L., Iera, A. and Morabito, G., 2010. The internet of things: A survey. *Computer networks*, 54(15), pp.2787-2805.
- Badran, O., Mekhilef, S., Mokhlis, H. and Dahalan, W., 2017. Optimal reconfiguration of distribution system connected with distributed generations: A review of different methodologies. *Renewable and Sustainable Energy Reviews*, 73, pp.854-867.
- Baker, A.D., 1998. A survey of factory control algorithms that can be implemented in a multi-agent heterarchy: dispatching, scheduling, and pull. *Journal of Manufacturing Systems*, 17(4), p 297.
- Banos, R., Manzano-Agugliaro, F., Montoya, F.G., Gil, C., Alcayde, A. and Gómez, J., 2011. Optimization methods applied to renewable and sustainable energy: A review. *Renewable and Sustainable Energy Reviews*, 15(4), pp.1753-1766.
- Bar-Yam, Y., 1997. Dynamics of complex systems (Vol. 213). Reading, MA. Addison-Wesley.
- Barbosa, J., Leitão, P., Adam, E. and Trentesaux, D., 2015. Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution. *Computers in Industry*, 66, pp.99-111.
- Barbosa, J., Leitão, P., Adam, E. and Trentesaux, D., 2012, March. Nervousness in Dynamic Self-organized Holonic Multi-agent Systems. In: *Highlights on Practical Applications of Agents and Multi-Agent Systems*, pp. 9–17. Springer, Berlin, Heidelberg (2012).
- Barbosa, J., Leitão, P. and Pereira, A.I., 2011, June. Combining adaptation and optimization in bio-inspired multi-agent manufacturing systems. In *Industrial Electronics (ISIE), 2011 IEEE International Symposium on* (pp. 1773-1778). IEEE.
- Batchkova, I., Popov, G., Karamishev, H. and Stambolov, G., 2013. Dynamic reconfigurability of control systems using IEC 61499 standard. *IFAC Proceedings Volumes*, 46(8), pp.256-261.
- Baykasoğlu, A. and Özbakır, L., 2008. Analysing the effect of flexibility on manufacturing systems performance. *Journal of Manufacturing Technology Management*, 19(2), pp.172-193.
- Bedford, D.S. and Malmi, T., 2015. Configurations of control: An exploratory analysis. *Management Accounting Research*, 27, pp.2-26.
- Bekkar, A., Guemri, O., Bekrar, A., Aissani, N., Beldjilali, B. and Trentesaux, D., 2016. An Iterative Greedy Insertion Technique for Flexible Job Shop Scheduling Problem. *IFAC-PapersOnLine*, 49(12), pp.1956-1961.

## References

---

- Belisario, L.S. and Pierreval, H., 2013, October. A conceptual framework for analyzing adaptable and reconfigurable manufacturing systems. In *Proceedings of the 5th International Conference on Industrial Engineering and Systems Management (IESM'13)*, pp.644–650, Rabat, Maroc, 28-30 octobre 2013. IEEE.
- Bernon, C., Gleizes, M.P. and Picard, G., 2007, September. Enhancing self-organising emergent systems design with simulation. In *International Workshop on Engineering Societies in the Agents World* (pp. 284-299). Springer, Berlin, Heidelberg.
- Blum, C. and Roli, A., 2003. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3), pp.268-308.
- Bondalapati, K. and Prasanna, V.K., 2002. Reconfigurable computing systems. *Proceedings of the IEEE*, 90(7), pp.1201-1217.
- Borangiu, T., Raileanu, S., Trentesaux, D., Berger, T. and Iacob, I., 2013. Distributed manufacturing control with extended CNP interaction of intelligent products. *Journal of Intelligent Manufacturing*, 25(5), pp.1065-1075.
- Bordoloi, S.K., Cooper, W.W. and Matsuo, H., 2009. Flexibility, adaptability, and efficiency in manufacturing systems. *Production and Operations Management*, 8(2), pp.133-150.
- Botti, V. and Giret, A., 2008. *ANEMONA: A multi-agent methodology for Holonic Manufacturing Systems*. Springer Science & Business Media.
- Bouazza, W., Sallez, Y., Aissani, N. and Beldjilali, B., 2015. A Model for Manufacturing Scheduling Optimization through Learning Intelligent Products. In: *Borangiu T., Thomas A., Trentesaux D. (eds) Service Orientation in Holonic and Multi-agent Manufacturing. Studies in Computational Intelligence*, vol 594. pp. 233-241.
- Brambilla, M., Ferrante, E., Birattari, M. and Dorigo, M., 2013. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1), pp.1-41.
- Brdys, M.A., Grochowski, M., Gminski, T., Konarczak, K. and Drewa, M., 2008. Hierarchical predictive control of integrated wastewater treatment systems. *Control Engineering Practice*, 16(6), pp.751-767.
- Brennan, R.W., 2016. Distributed Intelligent Sensing and Control for Manufacturing Automation. *Distributed Control Applications: Guidelines, Design Patterns, and Application Examples with the IEC 61499*, 9, p.317.
- Brennan, R.W. and Norrie, D.H., 2001. Agents, holons and function blocks: distributed intelligent control in manufacturing. *Journal of Applied Systems Studies*, 2(1), pp.1-19.
- Brennan, R.W., Zhang, X., Xu, Y. and Norrie, D.H., 2002. A reconfigurable concurrent function block model and its implementation in real-time Java. *Integrated Computer-Aided Engineering*, 9(3), pp.263-279.
- Brennan, R.W., Vrba, P., Tichy, P., Zoitl, A., Sünder, C., Strasser, T. and Marik, V., 2008. Developments in dynamic and intelligent reconfiguration of industrial automation. *Computers in Industry*, 59(6), pp.533-547.
- Brennan, R.W., Fletcher, M. and Norrie, D.H., 2002. An agent-based approach to reconfiguration of

## References

---

- real-time distributed control systems. *IEEE transactions on Robotics and Automation*, 18(4), pp.444-451.
- Broman, G.I. and Robèrt, K.H., 2017. A framework for strategic sustainable development. *Journal of Cleaner Production*, 140, pp.17-31.
- Browne, J., Dubois, D., Rathmill, K., Sethi, S.P. and Stecke, K.E., 1984. Classification of flexible manufacturing systems. *The FMS magazine*, 2(2), pp.114-117.
- Brusaferrì, A., Ballarino, A., Cavadini, F.A., Manzocchi, D. and Mazzolini, M., 2014, September. CPS-based hierarchical and self-similar automation architecture for the control and verification of reconfigurable manufacturing systems. In *Emerging Technology and Factory Automation (ETFA), 2014 IEEE* (pp. 1-8). IEEE.
- Cardin, O., Trentesaux, D., Thomas, A., Castagna, P., Berger, T. and El-Haouzi, H.B., 2015. Coupling predictive scheduling and reactive control in manufacturing hybrid control architectures: state of the art and future challenges. *Journal of Intelligent Manufacturing*, 28(7), 1503-1517.
- Cardin, O. and Castagna, P., 2009. Using online simulation in Holonic manufacturing systems. *Engineering Applications of Artificial Intelligence*, 22(7), pp.1025-1033.
- Chaari, T., Chaabane, S., Aissani, N. and Trentesaux, D., 2014, May. Scheduling under uncertainty: Survey and research directions. In *Advanced Logistics and Transport (ICALT), 2014 International Conference on* (pp. 229-234). IEEE.
- Chryssolouris, G., Efthymiou, K., Papakostas, N., Mourtzis, D. and Pagoropoulos, A., 2013. Flexibility and complexity: is it a trade-off?. *International Journal of Production Research*, 51(23-24), pp.6788-6802.
- Chu, Y., You, F. and Wassick, J.M., 2014. Hybrid method integrating agent-based modeling and heuristic tree search for scheduling of complex batch processes. *Computers & Chemical Engineering*, 60, pp.277-296.
- Chunjie, Z., Chunjie, X., Hui, C. and Huajing, F., 2007. Genetic algorithm-based dynamic reconfiguration for networked control system. *Neural computing and applications*, 17, pp.153-160.
- Cimino, D., Ferrero, A., Queirolo, L., Bellotti, F., Berta, R. and De Gloria, A., 2017, September. A Low-Cost, Open-Source Cyber Physical System for Automated, Remotely Controlled Precision Agriculture. In *International Conference on Applications in Electronics Pervading Industry, Environment and Society* (pp. 191-203). Springer, Cham.
- Cotta, C., Sevaux, M. and Sørensen, K. eds., 2008. *Adaptive and multilevel metaheuristics* (Vol. 136). Springer.
- Da Silva, R.M., Junqueira, F., Santos Filho, D.J. and Miyagi, P.E., 2016. Control architecture and design method of reconfigurable manufacturing systems. *Control Engineering Practice*, 49, pp.87-100.
- Da Silva, R.M., Blos, M.F., Junqueira, F., dos Santos Filho, D.J. and Miyagi, P.E., 2014. Control architecture for reconfigurable and distributed manufacturing system. In *Proceedings of ABCM Symposium Series in Mechatronics* (Vol. 6, pp. 219-230).
- De Toni, A. and Tonchia, S., 1998. Manufacturing flexibility: a literature review. *International journal*

## References

---

- of production research*, 36(6), pp.1587-1617.
- Debenham, J. and Prodan, A., 2013, August. The ‘Two-Hop Principle’: Stability of MAS in Communications Networks. In *International Conference on Industrial Applications of Holonic and Multi-Agent Systems* (pp. 281-292). Springer, Berlin, Heidelberg.
- Demesure, G., Defoort, M., Bekrar, A., Trentesaux, D. and Djemaï, M., 2016. Navigation Scheme with Priority-Based Scheduling of Mobile Agents: Application to AGV-Based Flexible Manufacturing System. *Journal of Intelligent & Robotic Systems*, 82(3-4), p.495.
- Demir, Y. and İşleyen, S.K., 2013. Evaluation of mathematical models for flexible job-shop scheduling problems. *Applied Mathematical Modelling*, 37(3), pp.977-988.
- Dennis, L.A., Fisher, M., Aitken, J.M., Veres, S.M., Gao, Y., Shaukat, A. and Burroughes, G., 2014. Reconfigurable autonomy. *KI-Künstliche Intelligenz*, 28(3), pp.199-207.
- Detzler, S., Eichhorn, C. and Karnouskos, S., 2015, September. Charging optimization of enterprise electric vehicles for participation in demand response. In *Smart Electric Distribution Systems and Technologies (EDST), 2015 International Symposium on* (pp. 284-289). IEEE.
- Dias-Ferreira, J., Ribeiro, L., Akillioglu, H., Neves, P. and Onori, M., 2016. BIOSOARM: a bio-inspired self-organising architecture for manufacturing cyber-physical shopfloors. *Journal of Intelligent Manufacturing*, pp.1-24.
- Dias-Ferreira, J., Ribeiro, L., Akillioglu, H., Neves, P., Maffei, A. and Onori, M., 2014, July. Characterization of an agile bio-inspired shop-floor. In *Industrial Informatics (INDIN), 2014 12th IEEE International Conference on*(pp. 404-410). IEEE.
- Dilts, D.M., Boyd, N.P. and Whorms, H.H., 1991. The evolution of control architectures for automated manufacturing systems. *Journal of manufacturing systems*, 10(1), pp.79-93.
- Di Marzo Serugendo, G., Gleizes, M.P. and Karageorgos, A., 2006. Self-organisation and emergence in MAS: An overview. *Informatica*, 30(1).
- Domingos, P., 2012. A few useful things to know about machine learning. *Communications of the ACM*, 55(10), pp.78-87.
- Duffie, N.A., 1990. Synthesis of heterarchical manufacturing systems. *Computers in Industry*, 14(1-3), pp.167-174.
- Eddahech, A., Chtourou, S. and Chtourou, M., 2013. Hierarchical neural networks based prediction and control of dynamic reconfiguration for multilevel embedded systems. *Journal of Systems Architecture*, 59(1), pp.48-59.
- Eisenring, M. and Platzner, M., 2002. A framework for run-time reconfigurable systems. *The Journal of Supercomputing*, 21(2), pp.145-159.
- ElMaraghy, W., ElMaraghy, H., Tomiyama, T. and Monostori, L., 2012. Complexity in engineering design and manufacturing. *CIRP Annals-Manufacturing Technology*, 61(2), pp.793-814.
- ElMaraghy, H.A., 2008. Changing and evolving products and systems—models and enablers. In *Changeable and reconfigurable manufacturing systems* (pp. 25-45). Springer London.
- ElMaraghy, H.A., 2006. Flexible and reconfigurable manufacturing systems paradigms. *International*

## References

---

- journal of flexible manufacturing systems*, 17(4), pp.261-276.
- Estrin, G., 1960, May. Organization of computer systems: the fixed plus variable structure computer. In *Papers presented at the May 3-5, 1960, western joint IRE-AIEE-ACM computer conference* (pp. 33-40). ACM.
- Filip, F.G., 2008. Decision support and control for large-scale complex systems. *Annual Reviews in Control*, 32(1), pp.61-70.
- Frei, R., Barata, J. and Onori, M., 2007, June. Evolvable production systems context and implications. In *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on* (pp. 3233-3238). IEEE.
- Gandomi, A. and Haider, M., 2015. Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2), pp.137-144.
- Giret, A., Trentesaux, D. and Prabhu, V., 2015. Sustainability in manufacturing operations scheduling: A state of the art review. *Journal of Manufacturing Systems*, 37, pp.126-140.
- Graves, S.C. and Willems, S.P., 2005. Optimizing the supply chain configuration for new products. *Management science*, 51(8), pp.1165-1180.
- Gubbi, J., Buyya, R., Marusic, S. and Palaniswami, M., 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7), pp.1645-1660.
- Gumzej, R., Colnarič, M. and Halang, W.A., 2009. A reconfiguration pattern for distributed embedded systems. *Software and systems modeling*, 8(1), pp.145-161.
- Gunasekaran, A. and Ngai, E.W., 2012. The future of operations management: an outlook and analysis. *International Journal of Production Economics*, 135(2), pp.687-701.
- Gungor, V.C. and Lambert, F.C., 2006. A survey on communication networks for electric system automation. *Computer Networks*, 50(7), pp.877-897.
- Gurgen, L., Gunalp, O., Benazzouz, Y. and Gallissot, M., 2013, March. Self-aware cyber-physical systems and applications in smart buildings and cities. In *Proceedings of the Conference on Design, Automation and Test in Europe* (pp. 1149-1154). EDA Consortium.
- Hacking, T. and Guthrie, P., 2008. A framework for clarifying the meaning of Triple Bottom-Line, Integrated, and Sustainability Assessment. *Environmental Impact Assessment Review*, 28(2), pp.73-89.
- Hamieh, A., Deneux, D. and Tahon, C., 2017. BiMov: BIM-Based Indoor Path Planning. In *Advances on Mechanics, Design Engineering and Manufacturing* (pp. 889-899). Springer International Publishing.
- Hashemi, S., Monfaredi, K. and Masdari, M., 2013. Using cloud computing for e-government: challenges and benefits. *International Journal of Computer, Information, Systems and Control Engineering*, 7(9), pp.596-603.
- Hegde, C. and Gray, K.E., 2017. Use of machine learning and data analytics to increase drilling efficiency for nearby wells. *Journal of Natural Gas Science and Engineering*, 40, pp.327-335.

## References

---

- Herrmann, C., Schmidt, C., Kurle, D., Blume, S. and Thiede, S., 2014. Sustainability in Manufacturing and Factories of the Future. *International Journal of precision engineering and manufacturing-green technology*, 1(4), pp.283-292.
- Hjelmås, E. and Low, B.K., 2001. Face detection: A survey. *Computer vision and image understanding*, 83(3), pp.236-274.
- Holvoet, T. and Valckenaers, P., 2007, May. Exploiting the environment for coordinating agent intentions. In *International Workshop on Environments for Multi-Agent Systems* (pp. 51-66). Springer Berlin Heidelberg.
- Jamshidi, M., Betancourt, A.J. and Gomez, J., 2011. Cyber-physical control of unmanned aerial vehicles. *Scientia Iranica*, 18(3), pp.663-668.
- Jelali, I.M., 2013. Introduction. In *Control Performance Management in Industrial Automation* (pp. 1-25). Springer London.
- Jennings, N.R., 2000. On agent-based software engineering. *Artificial intelligence*, 117(2), pp.277-296.
- Jiang, J., 1994. Design of reconfigurable control systems using eigenstructure assignments. *International Journal of Control*, 59(2), pp.395-410.
- Jiang, J. and Yu, X., 2012. Fault-tolerant control systems: A comparative study between active and passive approaches. *Annual Reviews in control*, 36(1), pp.60-72.
- Jimenez, J.F., Bekrar, A., Zambrano-Rey, G., Trentesaux, D. and Leitão, P., 2017a. Pollux: a dynamic hybrid control architecture for flexible job shop systems. *International Journal of Production Research*, 55(15), pp.4229-4247.
- Jimenez J.F., Zambrano-Rey G., Bekrar A., Trentesaux D., Leitão P. 2017b Analysing the Impact of Rescheduling Time in Hybrid Manufacturing Control. In: *Borangiu T., Trentesaux D., Thomas A., Leitão P., Oliveira J. (eds) Service Orientation in Holonic and Multi-Agent Manufacturing. Studies in Computational Intelligence*, vol 694. Pp 225 – 236.
- Jimenez, J.F., Bekrar, A., Giret, A., Leitão, P. and Trentesaux, D., 2016. A dynamic hybrid control architecture for sustainable manufacturing control. *IFAC-PapersOnLine*, 49(31), pp.114-119.
- Jimenez, J.F., Bekrar, A., Trentesaux, D. and Leitão, P., 2016. A switching mechanism framework for optimal coupling of predictive scheduling and reactive control in manufacturing hybrid control architectures. *International Journal of Production Research*, 54(23), pp.7027-7042.
- Jimenez, J.F., Bekrar, A., Trentesaux, D., Rey, G.Z. and Leitão, P., 2015a. Governance mechanism in control architectures for flexible manufacturing systems. *IFAC-PapersOnLine*, 48(3), pp.1093-1098.
- Jimenez, J.F., Bekrar, A., Trentesaux, D. and Leitão, P., 2015b, September. An Approach for Characterizing the Operating Modes in Dynamic Hybrid Control Architectures. In *International Conference on Industrial Applications of Holonic and Multi-Agent Systems* (pp. 108-119). Springer, Cham.
- Jimenez, J.F., Bekrar, A., Trentesaux, D., Montoya-Torres, J.R. and Leitão, P., 2013, October. State of the art and future trends of optimality and adaptability articulated mechanisms for manufacturing

## References

---

- control systems. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*. pp. 1265-1270. IEEE.
- Kaelbling, L.P., 1987. An architecture for intelligent reactive systems. In *Morgan Kaufmann, Proceedings of the 1986 Workshop: Reasoning about Actions and Plans, Editors: Georgeff, M, Lansky, A*, vol. 30, pp. 395-410).
- Kagermann, H., Helbig, J., Hellinger, A. and Wahlster, W., 2013. *Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry; final report of the Industrie 4.0 Working Group*. Forschungsunion.
- Kale, M.M. and Chipperfield, A.J., 2005. Stabilized MPC formulations for robust reconfigurable flight control. *Control Engineering Practice*, 13(6), pp.771-788.
- Kasinger, H., 2010. *Design and Operation of Efficient Self-Organizing Systems* (Doctoral dissertation, University of Augsburg).
- Katragjini, K., Vallada, E. and Ruiz, R., 2015, October. Rescheduling flowshops under simultaneous disruptions. In *Industrial Engineering and Systems Management (IESM), 2015 International Conference on* (pp. 84-91). IEEE.
- Khalgui, M., Mosbahi, O., Li, Z. and Hanisch, H.M., 2011. Reconfiguration of distributed embedded-control systems. *IEEE/ASME Transactions on Mechatronics*, 16(4), pp.684-694.
- Konstantopoulos, I.K. and Antsaklis, P.J., 1999. An optimization approach to control reconfiguration. *Dynamics and Control*, 9(3), pp.255-270.
- Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G. and Van Brussel, H., 1999. Reconfigurable manufacturing systems. *CIRP Annals-Manufacturing Technology*, 48(2), pp.527-540.
- Koren, Y., Hu, S.J. and Weber, T.W., 1998. Impact of manufacturing system configuration on performance. *CIRP Annals-Manufacturing Technology*, 47(1), pp.369-372.
- Kortuem, G., Kawsar, F., Sundramoorthy, V. and Fitton, D., 2010. Smart objects as building blocks for the internet of things. *IEEE Internet Computing*, 14(1), pp.44-51.
- Kumar, V., Kumar, S., Tiwari, M.K. and Chan, F., 2008. Performance evaluation of flexible manufacturing systems under uncertain and dynamic situations. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 222(7), pp.915-934.
- Lee, E.A., 2015. The past, present and future of cyber-physical systems: A focus on models. *Sensors*, 15(3), pp.4837-4869.
- Lee, I. and Lee, K., 2015. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4), pp.431-440.
- Leitão, P., Colombo, A.W. and Karnouskos, S., 2016. Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges. *Computers in Industry*, 81, pp.11-25.
- Leitão, P. and Restivo, F., 2006. ADACOR: A holonic architecture for agile and adaptive manufacturing control. *Computers in industry*, 57(2), pp.121-130.

## References

---

- Lincoln, N., Veres, S.M., Dennis, L.A., Fisher, M. and Lisitsa, A., 2010, August. An agent based framework for adaptive control and decision making of autonomous vehicles. In *ALCOSP* (pp. 310-317).
- Liu, C., Hildre, H.P., Zhang, H. and Rølvåg, T., 2016. Product architecture design of multi-modal products. *Research in Engineering Design*, 27(4), pp.331-346.
- Liu, C.H. and Zhang, Y. eds., 2015. *Cyber Physical Systems: Architectures, Protocols and Applications* (Vol. 22). CRC Press.
- Luke, S., 2009. *Essentials of metaheuristics* (Vol. 113). Raleigh: Lulu.
- Lyke, J.C., Christodoulou, C.G., Vera, G.A. and Edwards, A.H., 2015. An introduction to reconfigurable systems. *Proceedings of the IEEE*, 103(3), pp.291-317.
- Madureira, A., Ramos, C. and Silva, S.D.C., 2001, July. A genetic approach for dynamic job-shop scheduling problems. In *4th MetaHeuristics International Conference*.
- 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*.
- Manceaux, A. (2015). Contribution au rééquilibrage dynamique des lignes d'assemblage: modélisation, résolutions et applications (Doctoral dissertation, Université de Lorraine).
- Martin, C. and Barber, K.S., 2006. Adaptive decision-making frameworks for dynamic multi-agent organizational change. *Autonomous Agents and Multi-Agent Systems*, 13(3), pp.391-428.
- Marzouki, B. and Driss, O.B., 2016. Multi Agent Model Based on Chemical Reaction Optimization for Flexible Job Shop Problem. In *Computational Collective Intelligence* (pp. 29-38). Springer, Cham.
- Medvidovic, N., 1996, October. ADLs and dynamic architecture changes. In *Joint proceedings of the second international software architecture workshop (ISAW-2) and international workshop on multiple perspectives in software development (Viewpoints' 96) on SIGSOFT'96 workshops* (pp. 24-27). ACM.
- Mell, P. and Grance, T., 2011. The NIST definition of cloud computing. <https://www.nist.gov>
- Merriam-Webster, 2004. *Merriam-Webster's collegiate dictionary*. Merriam-Webster.
- Michalos, G., Makris, S., Aivaliotis, P., Matthaikakis, S., Sardelis, A. and Chryssolouris, G., 2015. Autonomous production systems using open architectures and mobile robotic structures. *Procedia CIRP*, 28, pp.119-124.
- Mircea, M. and Andreescu, A.I., 2011. Using cloud computing in higher education: A strategy to improve agility in the current financial crisis. *Communications of the IBIMA*.
- Mohamed, M., Murray, A. and Mohamed, M., 2010. The role of information and communication technology (ICT) in mobilization of sustainable development knowledge: a quantitative evaluation. *Journal of Knowledge Management*, 14(5), pp.744-758.
- Monostori, L., Valckenaers, P., Dolgui, A., Panetto, H., Brdys, M. and Csáji, B.C., 2015. Cooperative

## References

---

- control in production and logistics. *Annual Reviews in Control*, 39, pp.12-29.
- Montoya-Torres, J.R., 2015, May. Designing sustainable supply chains based on the triple bottom line approach. In *Advanced Logistics and Transport (ICALT), 2015 4th International Conference on* (pp. 1-6). IEEE.
- Morariu, O., Borangiu, T., Morariu, C. and Răileanu, S., 2016, May. Service Oriented Mechanisms for Smart Resource Allocation in Private Manufacturing Clouds. In *International Conference on Exploring Services Science* (pp. 367-383). Springer, Cham.
- Mouelhi-Chibani, W., Pierreval, H., 2010. Training a neural network to select dispatching rules in real time. *Computers & Industrial Engineering*, 58(2), 249–256.
- Moutassem, W. and Anders, G.J., 2010. Configuration optimization of underground cables for best ampacity. *IEEE Transactions on Power Delivery*, 25(4), pp.2037-2045.
- Mouzon, G., Yildirim, M.B. and Twomey, J., 2007. Operational methods for minimization of energy consumption of manufacturing equipment. *International Journal of Production Research*, 45(18-19), pp.4247-4271.
- Neungmacha, W., Sethanan, K., Gen, M. and Theerakulpisut, S., 2013. Adaptive genetic algorithm for solving sugarcane loading stations with multi-facility services problem. *Computers and electronics in agriculture*, 98, pp.85-99.
- Neves, P., Ferreira, J., Onori, M. and Barata, J., 2011, November. Context and implications of learning in Evolvable Production Systems. In *IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society* (pp. 2740-2745). IEEE.
- Nof, S.Y., 2009. Automation: What it means to us around the world. In *Springer Handbook of Automation* (pp. 13-52). Springer Berlin Heidelberg.
- Novas, J.M., Van Belle, J., Saint Germain, B. and Valckenaers, P., 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*, 3–17. Berlin Heidelberg: Springer
- Onori, M., Barata, J. and Frei, R., 2006. Evolvable assembly systems basic principles. *Information Technology for Balanced Manufacturing Systems*, pp.317-328.
- Pach, C., Berger, T., Bonte, T. and Trentesaux, D., 2014. ORCA-FMS: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling. *Computers in Industry*, 65(4), pp.706-720.
- Paolucci, M., Kawamura, T., Payne, T.R. and Sycara, K., 2002, June. Semantic matching of web services capabilities. In *International Semantic Web Conference* (pp. 333-347). Springer Berlin Heidelberg.
- Phocas, M.C., Christoforou, E.G. and Matheou, M., 2015. Design, motion planning and control of a reconfigurable hybrid structure. *Engineering Structures*, 101, pp.376-385.
- Picard, G., Hübner, J.F., Boissier, O. and Gleizes, M.P., 2009, June. Reorganisation and self-organisation in multi-agent systems. In *1st International Workshop on Organizational Modeling*,

## References

---

- ORGMOD (pp. 66-80).
- Pierreval, H. and Paris, J.L., 2003. From 'simulation optimization' to 'simulation configuration' of systems. *Simulation Modelling Practice and Theory*, 11(1), pp.5-19.
- Pinedo, M., 2012. *Scheduling*. New York: Springer.
- Prabhu, V.V., Trentesaux, D. and Taisch, M., 2015. Energy-aware manufacturing operations. *International Journal of Production Research*, 53(23), 6994 -7004
- Quintanilla, F.G., Cardin, O., L'Anton, A. and Castagna, P., 2016. A modeling framework for manufacturing services in service-oriented holonic manufacturing systems. *Engineering Applications of Artificial Intelligence*, 55, pp.26-36.
- Rad, C.R., Hancu, O., Takacs, I.A. and Olteanu, G., 2015. Smart monitoring of potato crop: a cyber-physical system architecture model in the field of precision agriculture. *Agriculture and Agricultural Science Procedia*, 6, pp.73-79.
- Rada, R. and Holden, H., 2009. Online Education, Standardization, and Roles. In *Encyclopedia of Distance Learning, Second Edition* (pp. 1493-1496). IGI Global.
- Radziewska, S., 2016. Global Crisis, Financialization and Technological Development. *International Economic Policy*, (24), p.124.
- Raileanu, S., Parlea, M., Borangiu, T. and Stocklosa, O., 2012. A JADE Environment for Product Driven Automation of Holonic Manufacturing. In: *Borangiu T., Thomas A., Trentesaux D. (eds) Service Orientation in Holonic and Multi-Agent Manufacturing Control. Studies in Computational Intelligence*, vol 402, pp 265–277.
- Rardin, R.L. and Rardin, R.L., 1998. *Optimization in operations research* (Vol. 166). Upper Saddle River, NJ: Prentice Hall.
- Rivera, D., 2005, April. The effect of content customization on learnability and perceived workload. In *CHI'05 extended abstracts on Human factors in computing systems* (pp. 1749-1752). ACM.
- Rojas Ramírez, J., 1999. Evaluación de Alternativas ante Falla en Sistemas Flexibles de Producción. In *Duodécima Reunión de Verano de Potencia, Aplicaciones Industriales y Exposición Industrial*. Mexico: IEEE Mexico, pp. 1–6.
- Rooker, M.N., Ebenhofer, G. and Strasser, T., 2009, June. Reconfigurable control in distributed automation systems. In *Reconfigurable Mechanisms and Robots, 2009. ReMAR 2009. ASME/IFToMM International Conference on* (pp. 705-714). IEEE.
- Rosen, M.A. and Kishawy, H.A., 2012. Sustainable manufacturing and design: Concepts, practices and needs. *Sustainability*, 4(2), pp.154-174.
- Russell, S.J. and Norvig, P., 2002. *Artificial intelligence: a modern approach* (International Edition).
- Russom, P., 2011. Big data analytics. *TDWI best practices report, fourth quarter, 19*, p.40.
- Rutkowski, L., 2008. *Computational intelligence: methods and techniques*. Springer Science & Business Media.

## References

---

- Sanchez, L.E., Diaz-Pace, J.A., Zunino, A., Moisan, S. and Rigault, J.P., 2011. An approach based on feature models and quality criteria for adapting component-based systems. *Journal of Software Engineering Research and Development*, 3(1), p.10.
- Sayama, H., 2015. Introduction to the modeling and analysis of complex systems. Open SUNY Textbooks.
- Schirner, G., Erdogmus, D., Chowdhury, K., and Padir, T. (2013). The future of human-in-the-loop cyber-physical systems. *Computer*, 46(1), 36-45.
- Schwarz, J., Beloff, B. and Beaver, E., 2002. Use sustainability metrics to guide decision-making. *Chemical Engineering Progress*, 98(7), pp.58-63.
- Senehi, M.K. and Kramer, T.R., 1998. A framework for control architectures. *International Journal of Computer Integrated Manufacturing*, 11(4), pp.347-363.
- Seshia, S.A., Hu, S., Li, W. and Zhu, Q., 2016. Design Automation of Cyber-Physical Systems: Challenges, Advances, and Opportunities. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 36(9), pp.1421-1434.
- Setchi, R.M. and Lagos, N., 2004, June. Reconfigurability and reconfigurable manufacturing systems: state-of-the-art review. In *Industrial Informatics, 2004. INDIN'04. 2004 2nd IEEE International Conference on*. pp. 529-535. IEEE.
- Shah, P.A.A., Habib, M., Sajjad, T., Umar, M. and Babar, M., 2017, September. Applications and Challenges Faced by Internet of Things-A Survey. In *International Conference on Future Intelligent Vehicular Technologies*. pp. 182-188. Springer, Cham.
- Simon, G., Kovács házy, T. and Péceli, G., 2002. Transient management in reconfigurable control systems. *Technical Report*, Budapest University of Technology and Economics.
- Slack, N., 1983. Flexibility as a manufacturing objective. *International Journal of Operations & Production Management*, 3(3), pp.4-13.
- Sokolowski, J.A. and Banks, C.M. eds., 2012. *Handbook of real-world applications in modeling and simulation* (Vol. 2). John Wiley & Sons.
- Srini, V.P. and Shriver, B.D., 1984. A methodology for designing and modeling reconfigurable systems. *International Journal of Parallel Programming*, 13(5), pp.339-394.
- Stadtler, H. and Kilger, C., 2002. Supply Chain Management and Advanced Planning, 2000. *Ronald H. Ballou. Business Logistics Management*, pp.61-73.
- Szelke, E. and Kerr, R.M., 1994. Knowledge-based reactive scheduling. *Production Planning & Control*, 5(2), pp.124-145.
- Tan, Y. and Aufenanger, M., 2011, July. A real-time rescheduling heuristic using decentralized knowledge-based decisions for flexible flow shops with unrelated parallel machines. In *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on* (pp. 431-436). IEEE.
- Terkaj, W., Tolio, T. and Valente, A., 2009. A review on manufacturing flexibility. *Design of flexible production systems*, pp.41-61.

## References

---

- Thierer, A. and Castillo, A., 2016. Preparing for the Future of Artificial Intelligence. *Public Interest Comment*, Mercatus Center at George Mason University, Arlington, VA.
- Thomas, A., Trentesaux, D. and Valckenaers, P., 2012. Intelligent distributed production control. *Journal of Intelligent Manufacturing*, 20, 2507 – 2512.
- Thomas, A., El Haouzi, H., Klein, T., Belmokhtar, S. and Herrera, C., 2009. Architecture de systèmes contrôlés par la produit pour un environnement de juste à temps. *Journal Européen des Systèmes Automatisés (JESA)*, 43(4-5), pp.513-535.
- Trentesaux, D. and Millot, P., 2016. A Human-Centred Design to Break the Myth of the “Magic Human” in Intelligent Manufacturing Systems. In: *Borangiu T., Trentesaux D., Thomas A., McFarlane D. (eds) Service Orientation in Holonic and Multi-Agent Manufacturing. Studies in Computational Intelligence*, vol 640, pp. 103-113.
- Trentesaux, D. and Giret, A., 2015. Go-green manufacturing holons: a step towards sustainable manufacturing operations control. *Manufacturing Letters*, 5, pp.29-33.
- Trentesaux, D., Pach, C., Bekrar, A., Sallez, Y., Berger, T., Bonte, T., Leitão, P. and Barbosa, J., 2013. Benchmarking flexible job-shop scheduling and control systems. *Control Engineering Practice*, 21(9), pp.1204-1225.
- Trentesaux, D. and Prabhu, V.V., 2013. Introduction to Shop-Floor Control. *Wiley Encyclopedia of Operations Research and Management Science*.
- Trentesaux, D., and Rault, R. 2017a. Ethical behaviour of autonomous Non-Military Cyber-Physical systems. In *XIX Complex systems: control and modelling problems, International conference on*, pp. 26-34.
- Trentesaux, D., and Rault, R. 2017b. Designing Ethical Cyber-Physical Industrial Systems. *IFAC-PapersOnLine*, 50(1), 14934-14939.
- Trentesaux, D., 2009. Distributed control of production systems. *Engineering Applications of Artificial Intelligence*, 22(7), pp.971-978.
- Trentesaux, D., 2002. Pilotage hétérarchique des systèmes de production, In *Habilitation à diriger des recherches*, Université de Valenciennes et du Hainaut-Cambresis.
- Truyen, E., Vanhaute, B., Joosen, W., Verbaeten, P. and Jorgensen, B.N., 2001, April. A dynamic customization model for distributed component-based systems. In *Distributed Computing Systems Workshop, 2001 International Conference on* (pp. 147-152). IEEE.
- Ueda, K., Takenaka, T., Váncza, J. and Monostori, L., 2009. Value creation and decision-making in sustainable society. *CIRP Annals-Manufacturing Technology*, 58(2), pp.681-700.
- Valente, A., De Simone, L., Carpanzano, E. and Ferraris, F., 2012. A methodology for static and dynamic scheduling of automation tasks in reconfigurable production systems. *CIRP Journal of Manufacturing Science and Technology*, 5(4), pp.241-253.
- Valls, M.G. and Val, P.B., 2014. Comparative analysis of two different middleware approaches for reconfiguration of distributed real-time systems. *Journal of Systems Architecture*, 60(2), pp.221-233.
- Valls, M.G., López, I.R. and Villar, L.F., 2013. iLAND: An enhanced middleware for real-time

## References

---

- reconfiguration of service oriented distributed real-time systems. *IEEE Transactions on Industrial Informatics*, 9(1), pp.228-236.
- Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L. and Peeters, P., 1998. Reference architecture for holonic manufacturing systems: PROSA. *Computers in industry*, 37(3), pp.255-274.
- Vieira, G.E., Herrmann, J.W. and Lin, E., 2003. Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *Journal of scheduling*, 6(1), pp.39-62.
- Vlk, M. and Barták, R., 2015. Replanning in Predictive-reactive Scheduling. Association for the Advancement of Artificial Intelligence (www.aaai.org).
- Voß, S., 2001, August. Meta-heuristics: The state of the art. In *Workshop on Local Search for Planning and Scheduling* (pp. 1-23). Springer, Berlin, Heidelberg.
- Wallner, H.P., 1999. Towards sustainable development of industry: networking, complexity and eco-clusters. *Journal of Cleaner Production*, 7(1), pp.49-58.
- Walsh, J.D.A., Bordeleau, F. and Selic, B., 2007. Domain analysis of dynamic system reconfiguration. *Software & Systems Modeling*, 6(4), pp.355-380.
- Walsh, D., Bordeleau, F. and Selic, B., 2006, March. Change types of dynamic system reconfiguration. In *Engineering of Computer Based Systems, 2006. ECBS 2006. 13th Annual IEEE International Symposium and Workshop on* (pp. 10-19). IEEE.
- Wang, Y. and Hajli, N., 2017. Exploring the path to big data analytics success in healthcare. *Journal of Business Research*, 70, pp.287-299.
- Whitehead, B.M., Boschee, F. and Decker, R.H., 2012. *The principal: Leadership for a global society*. Sage Publications.
- Whitmore, A., Agarwal, A. and Da Xu, L., 2015. The Internet of Things—A survey of topics and trends. *Information Systems Frontiers*, 17(2), pp.261-274.
- Wilensky, U. and Evanston, I., 1999. NetLogo: Center for connected learning and computer-based modeling. *Northwestern University, Evanston, IL*, 4952.
- Xia, T., Xi, L., Pan, E. and Ni, J., 2016. Reconfiguration-oriented opportunistic maintenance policy for reconfigurable manufacturing systems. *Reliability Engineering & System Safety*, 166, pp.87-98.
- Xiang, J.H., Fan, L., Liu, K. and Wang, Z.K., 2010, March. Optimal Reconfiguration Control for Satellite Constellations with Fault Satellites Using Multiobjective Optimization Algorithm. In *Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference on* (Vol. 2, pp. 837-840). IEEE.
- Youssef, A.M. and ElMaraghy, H.A., 2007. Optimal configuration selection for reconfigurable manufacturing systems. *International Journal of Flexible Manufacturing Systems*, 19(2), pp.67-106.
- Yu, Y., Wang, X., Zhong, R.Y. and Huang, G.Q., 2016. E-commerce logistics in supply chain management: Practice perspective. *Procedia CIRP*, 52, pp.179-185.
- Zahmani, M.H., Atmani, B., Bekrar, A. and Aissani, N., 2015, May. Multiple priority dispatching rules for the job shop scheduling problem. In *Control, Engineering & Information Technology (CEIT)*,

## References

---

- 2015 3rd International Conference on (pp. 1-6). IEEE.
- Zambrano Rey, G., Bekrar, A., Trentesaux, D. and Zhou, B.H., 2015. Solving the flexible job-shop just-in-time scheduling problem with quadratic earliness and tardiness costs. *The International Journal of Advanced Manufacturing Technology*, 81(9-12), pp.1871-1891.
- Zambrano Rey, G., Bonte, T., Prabhu, V. and Trentesaux, D., 2014a. Reducing myopic behavior in FMS control: A semi-heterarchical simulation–optimization approach. *Simulation Modelling Practice and Theory*, 46, pp.53-75.
- Zambrano Rey, G., Bekrar, A., Prabhu, V. and Trentesaux, D., 2014b. Coupling a genetic algorithm with the distributed arrival-time control for the JIT dynamic scheduling of flexible job-shops. *International Journal of Production Research*, 52(12), pp.3688-3709.
- Zambrano Rey, G., Pach, C., Aissani, N., Bekrar, A., Berger, T. and Trentesaux, D., 2013. The control of myopic behavior in semi-heterarchical production systems: A holonic framework. *Engineering Applications of Artificial Intelligence*, 26(2), pp.800-817.
- Zhang, Y. and Jiang, J., 2008. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual reviews in control*, 32(2), pp.229-252.

# APPENDICES

# Appendix A

## Optimization-based principles

The term optimization, which comes generally from the operational research domain, has been used to describe the selection of the best solution of a set of alternatives, regarding to a specific quality or objective function. This term has been widely used in engineering, computer science, mathematics and economics. An optimality model, or optimization model serves as a decision-making tool or a problem-solving technique towards the selection of the best alternative from a set of feasible solutions. From the system perspective, the optimization is the process of improving the design, management and performance of systems typically choosing the best organisation of physical resources, people and information (Royston, 2013). Then, the concept of optimization sets the guidelines to synchronize the system in order to improve the performance and results of the system execution.

In the literature, there are several optimization models to be implemented, such as mathematical programming, combinatorial algorithms, network optimization, evolutionary algorithms, along many other. Either the model used, the general optimization process is a class of algorithms and techniques which employ certain characteristics to find optimal (or as optimal as possible) solutions to hard problems (Sean, 2010). In this sense, the optimization model used requires the specification in five basic characteristics: a) the representation of alternatives or solutions, b) a function that assess the quality of the alternatives, c) an iterative process guiding the improvement process, d) a stop-criteria to terminate the algorithm subject to a constraint, and e) choosing the best alternative or solution selection. In this dissertation, the following characteristics of the optimization model will be referred as the optimization-based principles.

- a) ***Representation of alternatives:*** As previously stated, the optimization model is a problem-solving technique that search the best solution from a set of alternatives. An alternative is defined as a set of possible assignments of available resources to pending tasks (Chryssolouris, Pierce, & Dicke, 1991). The modelling process need to characterize the possible solution through a standard representation. The *representation* is generally understood to mean a data structure used to define an individual, such a vector or data-tree (Sean, 2010). The representation of each solution is a portrait that contain the specification, configuration or decision to set up each of the actions regarding the problem resolution. In brief, it aims to characterize each alternative or solution,

categorize the alternative according to the representation data, serve as guideline in the problem resolution, and permits to evaluate a priori the expected results in the problem resolution.

- b) **Evaluation of alternatives:** this characteristic is a step that conjointly with the iterative process, assesses the alternatives regarding to a corresponding result when it is evaluated with objective function. The purpose of the objective function is to rank order all feasible alternative or choices from the system problem (Hazelrigg, 1998). Then, it permits to compare the alternatives consistent to the results of the objective function rather to the representation characteristics. In brief, it encompasses the determination of the consequences of the different alternatives with respect to the different criteria (Chryssolouris, Pierce, & Dicke, 1991). The evaluation or assessment can be performed by different methods, such as simulations, mathematical functions, fitness functions or multi-criteria functions. Nonetheless, the used evaluation of alternatives is related to the type of optimization model and the constraints from the problem resolution.
  
- c) **Iterative search process:** this process is an algorithm that takes a problem as input and returns a feasible solution to such problem, usually after evaluating several possible alternatives (Corral & Almendros-Jiménez, 2007). Besides of searching iteratively a best solution per some specific criteria, it encompasses a search strategy that decide the how to explore the feasible alternatives (Zhang & Korf, 1995). The iterative search process contains two main strategies (Blum & Roli, 2003): exploration (or diversification) and exploitation (or intensification). The term exploration refers to the characteristic of the search process that seeks a new feasible solution from the global set of alternatives. The term exploitation refers to the characteristic of the search process that seeks a new feasible solution in a focused region adjacent to an already proven quality alternative. The iterative search process may focus in either strategy or search in a mixed search, certainly chosen according to the objectives of the optimization model.
  
- d) **Stopping-criteria:** this characteristic of the optimization model is related to the termination condition of the iterative search process. Since this process computes successive approximations to the solution, a practical test is needed to determine when to terminate the iterations (Berry, Chan, & Demmel, 1987). There are several conditions for the stopping-criteria, such as execution time, a satisficing solution, number of iterations, or percentage gap with a pre-solved solution. Nevertheless, this stopping criteria must be aligned with the constraint with the optimization model.
  
- e) **Criteria for solution selection:** Optimization models can be classified between single-state or population-based techniques (Sean, 2010). While the single-based techniques focus in improving a single candidate during the iterative search process, the population-based techniques improve

conjointly a set of candidates (e.g. subset from the set of alternatives) during the process resulting into multiple improved solutions. For the solution selection, the single-state method is straightforward because in the only solution candidate. However, the population-based technique contains different candidates that are useful at the decision-taking, depending the needs during the problem solving. Each candidate from the population have distinct properties and strengths that cover diverse aspects and requirements during the decision-taking (Wang, Du, & Ding, 2011). Then, the selection criteria for the population-based may be the best candidate, the average candidate or an evaluation with a second function, for example. At the end, the choice in the optimization must be guided by the need to reflect overall system objectives (Chryssolouris, Pierce, & Dicke, 1991).

In resume, these optimization-based principles are common and basic characteristics found the optimization techniques. These optimization-based principles can be found in both single-state (i.e. hill climbing, simulated annealing, tabu search, or iterated local search) and population-based (i.e. genetic algorithm, particle swarm optimization, ant colony, or scatter search) optimization models.

## Appendix B

### **Approaches that categorization of reconfigurable control architectures.**

#### Garcia-Valls and Basanta-Val approach (García Valls and Basanta Val 2014)

The authors of this approach categorized the reconfiguration of systems according to the internal and external characteristics of the system. The criteria for this categorization are: **A) Software engineering or performance-based schemes**, where the first refers to changes to the programming code at component-level for transitioning to a different state, while the second refers to the specific changes to the whole system to identified configurations in order to meet a specific functionality; **B) Non real-time or real-time schemes**, where the first refers to the capability of changing to optimal configuration due to it is unbounded to time, while the second refers to reconfigurations limited in time and centred in reactivity; **C) Centralized or distributed schemes**, where the first refers to a hierarchical structure of the system, while the second refers to the heterarchical structure of the system; **D) Closed or open schemes**, where the first refers to changes to states and transitions that are previously known, while the second refers to changes to unknown states and transitions so further considerations are needed; **E) Entity or system-level execution schemes**, where the first deals with the low-level details of the run-time execution deciding the required adjustments to their parameters, while the second deals with optimization techniques and search techniques in a system level; and, **F) Hardware and Software reconfiguration**; where the first refers to changes of devices and hardware, while the second refers to changes in modules and software.

#### **Reconfigurable autonomy approach (Dennis et al. 2014).**

The authors of this approach concerns a classification of reconfiguration according the architecture, applications and control system. It assumes a modular nature of the architecture and allows a range of changeability to each type of reconfiguration. The types of reconfiguration are related to **A) hardware devices, B) low-control functioning, C) agent-level, D) agent decision-making level, and E) components interactions**. The hardware kind refers to adding/removing devices to encourage change of functionalities. The low-control functioning kind refers to the protocols followed by the controllers of the system. The agent-level kind refers to the changes of the agent or entity characteristics, such in goals, plans, actions, etc. The agent decision-making kind refers to the changes in the functioning of the agents or entities as a whole. The components interactions refer to changes in the relation between the entire pairwise components, such as master-slave or collaborative relationships.

### **Walsh, Bordeleau and Selic approach (Walsh, Bordeleau, and Selic 2007)**

The authors of this approach defines the reconfigurability according the way the system evolve and adapts to new conditions. It classifies the reconfiguration according type of change (i.e. structural or behavioural), and the location of change (i.e. Inter-component or intra-components). The types of reconfigurations are: **A) Architectural change**, as it is an inter-component change that may be structural, behavioural, or both and it may feature topology and/or protocol changes; **B) Topology change**; as a structural inter-component kind that changes the topology of a system through component addition and/or removal and it may feature substitution or protocol changes; **C) Substitution change**, as a structural intra-component change that replaces a particular component by another within a given system topology; **D) Protocol Change**, as a behavioural inter-component kind that changes the control flow and/or data flow of components and it may feature a topology or interface change; **E) Interface change**, as a behavioural intra-component kind that changes the externally observable behaviour of a component and it may feature internal change of one or more components to realize the new behaviour; and **F) Internal change** an intra-component kind either behavioural, structural, or both that changes the internal implementation of a component.

### **Zhang and Jiang approach (Zhang and Jiang 2008).**

The authors of this approach identify the type of changes according to the strategy and mechanism for the reconfiguration. This approach is closer to the continuous type of control but it serves as a starting point for applying to a discrete-event kind. In this case, the types of reconfiguration mechanism may be used for **A) optimization, B) switching, C) matching, D) following and E) compensation**. The optimization type refers to the decisional techniques that searches an optimal arrangement of the physical variables in order to optimize its objectives (i.e. finite-horizon series or linear quadratic models). The switching type refers to the use of different components, parameters modules with defined characteristics that can be exchanged with components parameters or modules of the system in order to encourage a change of behaviour (i.e. Gain scheduling or linear parameter-varying control). The matching type refers to the inclusion of a specific matrix that regulates the system through a control law (i.e. PI System matrix or Eigenstructure assignment). The following type refers to the use of a model of behaviour in order to make the control system to behave the same way (i.e. implicit or explicit model following). The compensation type refers to the aggregation an additional dampening method to the reconfiguration mechanism to compensate the deviation suffered during the perturbation (i.e. additive or adaptive compensation).

## Appendix C

## Appendices

Year	Author	Title
1984	Srini, V. P. & Shriver, B. D.	A methodology for designing and modeling reconfigurable systems
1989	Antsaklis, P. J.; Passino, K.	Towards intelligent autonomous control systems: Architecture and fundamental issues
1989	Passino, K. M. & Antsaklis, P	On the optimal control of discrete event systems
1990	Pancholy, Ashish; Muradali, Fi	A hierarchical reconfiguration strategy for bus-based multiprocessors
1995	Pete, Andras; Pattipati, Krish	Structural reconfiguration and informal coordination in administrative organizations
1996	Alencar, P. S. C. & de Lucena,	A logical framework for evolving software systems
1996	A. S. Kunnathur; S. Sampath	Simulation Conference, 1996. Proceedings. Winter
1998	Chen, Jong-Chen	Problem Solving with a Perpetual Evolutionary Learning Architecture
1998	Dasgupta, Dipankar	Evolving Neuro-Controllers for a Dynamic System Using Structured Genetic Algorithms
2000	Shen, Weiming; Maturana, Franc	MetaMorph II: an agent-based architecture for distributed intelligent design and manufacturing
2001	Balasubramanian, Sivaram; Bren	An architecture for metamorphic control of holonic manufacturing systems
2001	Bellows, Peter & Hutchings, Br	Designing Run-Time Reconfigurable Systems with JHDL
2001	Heron, J. P.; Woods, R.; Sezer	Development of a Run-Time Reconfiguration System with Low Reconfiguration Overhead
2001	Medova, E. A. & Scott, J. E.	Evolving System Architectures for Multimedia Network Design
2001	Ünsal, Cem; Kiliççöte, Han	A Modular Self-Reconfigurable Bipartite Robotic System: Implementation and Motion Planning
2001	Wills, L.; Kannan, S.; Sander,	An open platform for reconfigurable control
2002	Apte, Neeraj & Zeid, Ibrahim	Evolution of transparent manufacturing: An architecture for a Java-based controller of a CIM cell
2002	Brennan, R. W.; Fletcher, M.	An agent-based approach to reconfiguration of real-time distributed control systems
2002	Eisenring, Michael & Platzner,	A Framework for Run-time Reconfigurable Systems
2002	Salomon, Ralf	Evolving Receptive-Field Controllers for Mobile Robots
2002	Znamirowski, Lech; Palusinski,	Optimization Technique for Dynamic Reconfiguration of Programmable Analog/Digital Arrays
2003	Aubin, Jean-Pierre	Regulation of the evolution of the architecture of a network by tensors operating on coalitions of actors
2003	Pellegrini, Marie-Claude & Riv	Component Management in a Dynamic Architecture
2003	Puttegowda, Kiran; Lehn, David	Context Switching in a Run-Time Reconfigurable System
2003	Xu, Wei; Wang, Shi-Gang; Wang,	Towards an Efficient Self-organizing Reconfiguration Method for Self-reconfigurable Robots
2004	Apvrille, L.; de Saqui-Sannes,	Verifying Service Continuity in a Dynamic Reconfiguration Procedure: Application to a Satellite System
2005	Drozeski, Saha, Vachtsevanos.	A Fault Detection and Reconfigurable Control Architecture for Unmanned Aerial Vehicles
2005	Jeffrey, C.; Cutajar, R.; Rich	The Integration of On-Line Monitoring and Reconfiguration Functions into a Safety Critical Automotive Electronic Control Unit
2005	Minghui Wang; Shugen Ma; Bin L	Task planning and behavior scheduling for a reconfigurable planetary robot system
2006	EIMaraghy, Hoda A.	Flexible and reconfigurable manufacturing systems paradigms
2006	Horling, Bryan; Lesser, Victor	The Soft Real-Time Agent Control Architecture
2006	Leitao, Restivo.	ADACOR: a holonic architecture for agile and adaptive manufacturing control
2006	Martin, Cheryl & Barber, K. Su	Adaptive decision-making frameworks for dynamic multi-agent organizational change
2006	Meng, Fanli; Tan, Dalong & Wan	Development of agent for reconfigurable assembly system with JADE
2006	Rawashdeh, O. A. & Lump, J. E	Run-Time Behavior of Ardea: A Dynamically Reconfigurable Distributed Embedded Control Architecture
2006	Walsh, D.; Bordeleau, F. & Sel	Change types of dynamic system reconfiguration

## Appendices

Year	Author	Title
2006	Youssef, Ayman M. A. & ElMarag	Assessment of manufacturing systems reconfiguration smoothness
2006	Zoitl, Alois; Sunder, Christop	Dynamic reconfiguration of distributed control applications with reconfiguration services based on IEC 61499
2007	Bossuet, Lilian; Gogniat, Guy	Communication-Oriented Design Space Exploration for Reconfigurable Architectures
2007	Cao, Yang; ElMaraghy, Hoda A.	Reconfigurable Control Structure for Robots in Assembly
2007	Chunjie, Zhou; Chunjie, Xiang;	Genetic algorithm-based dynamic reconfiguration for networked control system
2007	Morrison, Ron; Balasubramaniam	A framework for supporting dynamic systems co-evolution
2007	Patouni, Eleni; Gault, Sophie;	Advanced reconfiguration framework based on game theoretical techniques in autonomic communication systems
2007	Walsh, James D'Arcy; Bordeleau	Domain analysis of dynamic system reconfiguration
2007	Youssef, Ayman M. A. & ElMarag	Optimal configuration selection for Reconfigurable Manufacturing Systems
2007	Min Yu; Weimin Zhang & Klemm,	Multi-agent based reconfigurable manufacturing execution system
2007	Yu, Zhang, Klemm.	Multi-agent based reconfigurable manufacturing execution system
2008	Aprile, Walter A.; Ruffaldi, E	A dynamically reconfigurable stereoscopic/panoramic vision mobile robot head controlled from a virtual environment
2008	Chokshi, Nirav & McFarlane, Du	A distributed architecture for reconfigurable control of continuous process operations
2008	Ferrarini, Luca; Fogliazza, Gi	Metamodeling Techniques Applied to the Design of Reconfigurable Control Applications
2008	Lavis, Benjamin; Furukawa, Tom	Dynamic space reconfiguration for Bayesian search and tracking with moving targets
2008	Li, Jun; Dai, Xianzhong & Meng	Improved net rewriting system-based approach to model reconfiguration of reconfigurable manufacturing systems
2008	Murta, Leonardo G. P.; vanÂ de	Continuous and automated evolution of architecture-to-implementation traceability links
2008	Qu, Yang; Tiensyrja, Soinen.	Design Flow Instantiation for Run-Time Reconfigurable Systems: A Case Study
2008	Zhang, Youmin & Jiang, Jin	Bibliographical review on reconfigurable fault-tolerant control systems
2009	Gumzej, Roman, Colnaric, Halang.	A reconfiguration pattern for distributed embedded systems
2009	Lombaerts, Chu, Mulder.	Flight Control Reconfiguration based on a Modular Approach
2009	Sghairi, Manel; Aubert, Jean-J	Distributed and reconfigurable architecture for flight control system
2009	Guowu Wei & Dai, J. S.	Reconfigurable Mechanisms and Robots, 2009. ReMAR 2009. ASME/IFTToMM International Conference on
2009	Rooker, Ebenhofer, Strasser.	Reconfigurable control in distributed automation systems
2009	Holvoet, Weyns, Valckenaers.	Patterns of delegate MAS
2010	Hsieh, Fu-Shiung	Design of reconfiguration mechanism for holonic manufacturing systems based on formal models
2010	Järvenpää, Eeva	Factory Level Logistics and Control Aspects for Flexible and Reactive Microfactory Concept
2010	Kasper, Toscher.	Reconfigurable Controllers-A Mechatronic Systems Approach
2010	Levi, Paul & Kernbach, Serge	Adaptive Control Mechanisms
2010	Li, Xinyu; Zhang, Chaoyong; Ga	An agent-based approach for integrated process planning and scheduling
2010	Lincoln, Nicholas K.; Veres, S	An Agent Based Framework for Adaptive Control and Decision Making of Autonomous Vehicles
2010	De la Mata, Rodriguez	Accident prevention by control system reconfiguration
2010	Pasek	Sandbox for Development of Evolving Manufacturing Control Architectures
2010	Ramdhany, Grace, Coulson et al.	Dynamic deployment and reconfiguration of ad-hoc routing protocols
2010	Renna, Paolo	Capacity reconfiguration management in reconfigurable manufacturing systems
2010	Xiang, Jun-hua; Fan, Li; Liu,	Optimal Reconfiguration Control for Satellite Constellations with Fault Satellites Using Multiobjective Optimization Algorithm

## Appendices

Year	Author	Title
2010	Xuemei, Huang	Distributed and reconfigurable control of lower level machines in automatic production line
2011	Abdelkrim, Hedi; Ben Othman, S	Runtime Reconfigurable MPSoC architecture for control drive system
2011	Baird, R.; Jorgenson, N. & Gam	Self-adapting workflow reconfiguration
2011	Javadian, Nikbaksh; Aghajani,	A multi-objective integrated cellular manufacturing systems design with dynamic system reconfiguration
2011	Lee, Gwo Giun; Wang, Ming-Jiun	Reconfigurable Architecture for Deinterlacer based on Algorithm/Architecture Co-Design
2011	Li, Wei	Evaluating the impacts of dynamic reconfiguration on the QoS of running systems
2011	Milutinovic, Dragan; Glavonjic	Reconfigurable robotic machining system controlled and programmed in a machine tool manner
2011	Unver, H. Ozgur	System architectures enabling reconfigurable laboratory-automation systems
2011	Wang, Li; Zhao, Jun; Wang, Wei	Dynamic scheduling with production process reconfiguration for cold rolling line
2012	Ben Letaifa, Asma & Tabbane, S	Reconfiguration process and routing management for service platforms
2012	Canal, Carlos; Camara	Structural reconfiguration of systems under behavioral adaptation
2012	Garibay, Eber Espino; Lucio, D	An approach: FPGA based dynamically reconfigurable architecture to enable several scheme controls for power converters
2012	Kallel, Slim; Hadj Kacem, Moha	Modeling and enforcing invariants of dynamic software architectures
2012	LOUREIRO, Rui; MERZOUKI, Rochd	Structural reconfiguration conditions based on bond graph approach: application to an intelligent autonomous vehicle
2012	Mannava, Vishnuvardhan & Rames	Procedia Engineering A novel adaptive re-configuration compliance design pattern for autonomic computing systems
2012	Mariappan, Ramasamy & Karthike	Optimization technique for virtual topology reconfiguration for IP-over-WDM networks with QoS parameters
2012	Morariu, Cristina; Morariu, Oc	Manufacturing Service Bus Integration Model for Implementing Highly Flexible and Scalable Manufacturing Systems
2012	Reinhart, G.; Niehues, M. & Os	Adaptive, Location-based Shop Floor Control
2012	Raileanu, Parlea, Borangiu et al.	A JADE environment for product driven automation of holonic manufacturing
2012	Valente, Anna; De Simone, Luis	A methodology for static and dynamic scheduling of automation tasks in reconfigurable production systems
2012	Zhang, Ping; He, Qian; Feng, Z	Reconfiguration decision making in cognitive wireless network
2013	An, Xin; Rutten, Eric; Diguët,	Discrete Control for Reconfigurable FPGA-based Embedded Systems
2013	Batchkova, Idilia; Popov, Geor	Dynamic reconfigurability of control systems using IEC 61499 standard
2013	Duhem, Muller, Aubry et al.	Design space exploration for partially reconfigurable architectures in real-time systems
2013	Eddahech, Chtourou, Chtourou.	Hierarchical neural networks based prediction and control of dynamic reconfiguration for multilevel embedded systems
2013	Garcia Valls, Marisol; Lopez,	iLAND: An Enhanced Middleware for Real-Time Reconfiguration of Service Oriented Distributed Real-Time Systems
2013	Gyurka, B.; Gherman, B.; Vaida	Optimal control for reducing the energy consumption of a reconfigurable parallel robot
2013	Murar, Mircea & Brad, Stelian	Control Architecture for Plug-and-Play Intelligent Axes within Fast Reconfigurable Manufacturing Equipments
2013	Novas, Van Belle.	A collaborative framework between a scheduling system and a holonic manufacturing execution system
2013	Oh, Seungjin; Ryu, Kwangyeol &	Reconfiguration framework of a supply network based on flexibility strategies
2013	Ostroumov, Sergey; Tsiopoulos,	Formal approach to agent-based dynamic reconfiguration in Networks-On-Chip
2013	Trabelsi, Chiraz; Meftali, Sam	Decentralized control for dynamically reconfigurable FPGA systems
2014	Beck, Antonio Carlos Schneider	A transparent and adaptive reconfigurable system
2014	Brusaferrì, Alessandro; Ballar	CPS-based hierarchical and self-similar automation architecture for the control and verification of reconfigurable manufacturing systems
2014	Dennis, Louise A.; Fisher, Mic	Reconfigurable Autonomy
2014	Dias-Ferreira, Joao; Ribeiro,	Characterization of an agile bio-inspired shop-floor

## Appendices

Year	Author	Title
2014	Essers, M. S. & Vaneker, T. H.	Evaluating a Data Distribution Service System for Dynamic Manufacturing Environments: A Case Study
2014	Garcia, Basanta.	Comparative analysis of two different middleware approaches for reconfiguration of distributed real-time systems
2014	Hou, Feili	Graph-based optimal reconfiguration planning for self-reconfigurable robots
2014	Jatzkowski, Jan & Kleinjohann,	ScienceDirect Towards self-reconfiguration of real-time communication within Cyber-Physical Systems
2014	Michniewicz, Reinhart.	Cyber-physical Robotics - Automated Analysis, Programming and Configuration of Robot Cells based on Cyber-physical-systems
2014	Sedighizadeh, Mostafa; Dakhem,	Optimal reconfiguration and capacitor placement for power loss reduction of distribution system using improved binary particle swarm optimization
2014	da Silva, R. M.; Blos, M. F. &	Control Architecture for Reconfigurable and distributed manufacturing system.
2014	da Silva, Robson Marinho; Junq	A method to design a manufacturing control system considering flexible reconfiguration
2014	Streit, Alberto; Rosch, Susann	Redeployment of control software during runtime for modular automation systems taking real-time and distributed I/O into consideration
2014	Pach, Berger, Bonte et al.	ORCA-FMS: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling
2014	Yao, Zhiyuan & El-Farra, Nael	Performance-Based Sensor Reconfiguration for Fault-Tolerant Control of Uncertain Spatially Distributed Processes
2014	Zambrano Rey, Bonte, Prabhu et al.	Reducing myopic behavior in FMS control: a semi-heterarchical simulation – optimization approach
2015	Bakhouya, Mohamed & Gaber, Jaa	Approaches for engineering adaptive systems in ubiquitous and pervasive environments
2015	Barbosa, Leitao, Adam et al.	Dynamic self-organization in holonic multi-agent manufacturing systems: the ADACOR evolution
2015	Berthier, Nicolas; An, Xin & M	Towards Applying Logico-numerical Control to Dynamically Partially Reconfigurable Architectures
2015	Coutinho, Nuno; Figueira, Dani	A control framework for abstract multiparty transport
2015	Furno, Lidia; Nielsen, Mikkel	ScienceDirect Centralised versus Decentralised Control Reconfiguration for Collaborating Underwater Robots
2015	Indriago, Carlos; Cardin, Oliv	Performance evaluation of holonic-based online scheduling for a switch arrival system
2015	Liu, Cong; Hildre, Hans Petter	Conceptual design of multi-modal products
2015	Jimenez, Bekrar, Trentesaux et al.	Governance mechanism in control architectures for flexible manufacturing systems
2015	Michalos, G.; Makris, S.; Aiva	Autonomous Production Systems Using Open Architectures and Mobile Robotic Structures
2015	Monostori, Valckenaers, Dolgui et al.	Cooperative control in production and logistics
2015	Oliveira, Barbosa.	Self-adaptation by coordination-targeted reconfigurations
2015	Phocas, Marios C.; Christoforo	Design, motion planning and control of a reconfigurable hybrid structure
2015	Priego, Rafael; Schutz, Daniel	Reconfiguration architecture for updates of automation systems during operation
2015	Reis, Joao Gabriel; Wanner, Lu	A Framework for Dynamic Real-Time Reconfiguration
2015	Sanchez, L. Emiliano; Diaz-Pac	An approach based on feature models and quality criteria for adapting component-based systems
2015	Shariatzadeh, Farshid; Kumar,	Intelligent control algorithms for optimal reconfiguration of microgrid distribution system
2015	da Silva, Benitez-Pina, Blos et al.	Modeling of reconfigurable distributed manufacturing control systems
2015	Borangiu, Raileanu, Berger et al.	Switching mode control strategy in manufacturing execution systems
2015	Vey, Hugging, Bodenbug et al.	Control reconfiguration of physically interconnected systems by decentralized virtual actuators
2015	Xu, Xiaodong; Dai, Xun; Liu, Y	Energy efficiency optimization-oriented control plane and user plane adaptation with a frameless network architecture for 5G
2015	Zielinski, Cezary; Szykiewicz	Reconfigurable control architecture for exploratory robots
2016	Cala, A.; Foehr, M.; Rohrmus,	Towards industrial exploitation of innovative and harmonized production systems
2016	Carlini, E. M.; Giannuzzi, G.	A Decentralized and Proactive Architecture based on the Cyber Physical System Paradigm for Smart Transmission Grids Modelling, Monitoring and Control
2016	Chu, Jing & Guo, Jian	Decentralized autonomous planning of cluster reconfiguration for fractionated spacecraft

## Appendices

Year	Author	Title
2016	Dias-Ferreira, Ribeiro, Akillioglu et al.	BIOSOARM: a bio-inspired self-organising architecture for manufacturing cyber-physical shopfloors
2016	Essers, M. S. & Vaneker, T. H.	Design of a decentralized modular architecture for flexible and extensible production systems
2016	Evangelopoulos, Vasileios A.;	Optimal operation of smart distribution networks: A review of models, methods and future research
2016	Gallehdari, Zahra; Meskin, Nad	A Distributed Control Reconfiguration and Accommodation for Consensus Achievement of Multiagent Systems Subject to Actuator Faults
2016	Gerostathopoulos, Bures, Hnetyinka et al.	Self-adaptation in software-intensive cyber-physical systems: From system goals to architecture configurations
2016	Gorecky, Dominic; Weyer, Steph	Design and Instantiation of a Modular System Architecture for Smart Factories
2016	Jimenez, Bekrar, Giret et al.	A dynamic hybrid control architecture for sustainable manufacturing control
2016	Leitao, Barbosa	Dynamic Switching Mechanism to Support Self-organization in ADACOR Holonic Control System
2016	Lesi, Vuk; Jakovljevic, Zivana	Towards Plug-n-Play numerical control for Reconfigurable Manufacturing Systems
2016	Marseu E.; Kolberg, D.; Birte	Interdisciplinary Engineering Methodology for changeable Cyber-Physical Production Systems
2016	Mazza, Andrea; Chicco, Gianfra	Multi-Objective Distribution Network Reconfiguration Based on Pareto Front Ranking
2016	Michalos, George; Sipsas, Plat	Decision making logic for flexible assembly lines reconfiguration
2016	Morariu, Octavian; Raileanu, S	Redundancy and scalability for virtualized MES systems with programmable infrastructure
2016	Pham, Nam Khanh; Kumar, Akash;	Leakage aware resource management approach with machine learning optimization framework for partially reconfigurable architectures
2016	de S. Dutra, Diogo & Silva, Jo	Product-Service Architecture (PSA): toward a Service Engineering perspective in Industry 4.0
2016	da Silva, Junqueira, Filho et al.	Control architecture and design method of reconfigurable manufacturing systems
2016	Steinegger, Michael; Plaschka,	A framework for modular and distributable control of reconfigurable robotic systems
2016	Sun, Ling; Li, Yameng & Gao, J	Architecture and Application Research of Cooperative Intelligent Transport Systems
2016	Valente, A.	Reconfigurable industrial robots: A stochastic programming approach for designing and assembling robotic arms
2016	Vasconcelos, Vasconcelos, Endler.	Dynamic and coordinated software reconfiguration in distributed data stream systems
2016	Wang, Guo Xin; Huang, Si Han;	Reconfiguration schemes evaluation based on preference ranking of key characteristics of reconfigurable manufacturing systems
2016	Wenxia, L. I. U.; Shuya, N. I.	Optimal planning of battery energy storage considering reliability benefit and operation strategy in active distribution system
2016	Weyer, Stephan; Meyer, Torben;	Future Modeling and Simulation of CPS-based Factories: an Example from the Automotive Industry
2016	Xia, Xi, Pan et al.	Reconfiguration-oriented opportunistic maintenance policy for reconfigurable manufacturing systems
2016	Yoon, SooCheol & Suh, Suk-Hwan	Manufacturing Information Bus from the Perspective of Cyber Physical Manufacturing System (CPMS)
2016	Zhang, Zhao, Xu	Time-optimal control for formation reconfiguration of multiple unmanned aerial vehicles
2017	Abeywickrama, Dhaminda B.	A survey of autonomic computing methods in digital service ecosystems
2017	Badran, Mekhilef, Mokhlis et al.	Optimal reconfiguration of distribution system connected with distributed generations: A review of different methodologies
2017	Bhattacharyya, Anirban; Mokhov	An empirical comparison of formalisms for modelling and analysis of dynamic reconfiguration of dependable systems
2017	Broman	A framework for strategic sustainable development
2017	Puik, Erik; Telgen, Daniel; va	Assessment of reconfiguration schemes for Reconfigurable Manufacturing Systems based on resources and lead time

## Appendix D

Authors	Domain	Contribution
Srini and Shriver, 1984	Dynamic systems. <b>Airport Logistics</b>	design a methodology to manage the communication system in airports, which based on the principles of dataflow, it use undirected or partially directed dataflow graphs to have a dynamic structure for reconfiguration purposes. The system contains switches for the dataflow between nodes in order to turn on/off the communication according to some patterns from condition-action thresholds.
Pete et al., 1995	Optimize design of public organizations. <b>Administration</b>	propose a methodology to determine the structure and evolution of a public organization in order to jointly optimize the reliability of service and the strategy against institutional failures. The paper suggests a structural adaptation via selecting an appropriate serial-parallel arrangement of individual units by a graph formalism and an allocation of probabilities to the decision strategy.
Balasubramanian et al., 2001	Real time control <b>Manufacturing</b>	uses the function block standard (IEC 61499) to characterize a reactive behaviour of homogeneous agents for developing a metamorphic control system with a multi-sensor real-system. The reconfiguration of this approach is included by considering redundancy techniques for the agent behaviour and a selecting the best technique with a bidding mechanism.
Wills et al. 2001	UAV devices <b>Engineering</b>	describes an open control platform that coordinates distributed interaction among diverse components and supports dynamic reconfiguration and customization of the components in real time. The reconfiguration accommodates rapidly by changing applications and devices with new generic modules of these components via a coordination mechanism between redundancy modules.
Brennan et al., 2002	Real-time systems <b>Manufacturing</b>	develop a technique to achieve automatic reconfiguration that results in predictable and stable system behaviour in a real-time environment. The reconfiguration searches in parallel a better control architecture (using also redundancy function blocks) and change to the new configuration when is needed.
Einsering and Platner, 2002	Multi-FPGA <b>Electronics</b>	propose a methodology and a design representation of for run-time reconfiguration systems. A hierarchical component creates and select a number of configurations per component of the system (i.e. FPGA), where each configuration contains all the necessary tasks and buffers together with tasks responsible for reconfiguration control.
Pellegrini and Riveill, 2003	Dynamic architecture <b>Computer science</b>	propose the inclusion of a mechanism that manages the components in the dynamic architectures. The reconfiguration approach is based in a configurator that may modify the bidding between any two components in

		order to modify the interrelation between them. A new structure with an emerging behaviour is achieved with this approach.
Rawashdeh and Lumpp, 2006	UAV devices <b>Engineering</b>	address a framework for developing a dependable and hierarchical embedded systems based on graphical software specification and fault-tolerant techniques. A central component searches between different dependency graphs (used to specify interaction and interdependencies) to find a viable mapping of functional software modules.
Leitao and Restivo (2006)	Assembly lines <b>Manufacturing</b>	proposes a holonic architecture that introduces an adaptive control for improving agility and flexibility against unexpected perturbations. This approach uses pheromone-like spreading mechanism to reconfigure dynamically the quality of holon-interaction and encourage an emergence in the control architecture.
Bossuet et al., 2007	FPGA devices <b>Electronics</b>	propose a combination of algorithmic explorations towards a rapid definition of an efficient reconfigurable architecture for electronic devices (i.e. FPGA). The reconfiguration is based in an automatic estimation tool that computes and evaluates a performance indicator of the possible new architecture (usage rate of communication and processing resources).
Cao et al., 2007	Robot control <b>Robotics</b>	proposes to introduce a reconfiguration controller to meet the new on-line demands for changeability in robotic systems. This approach counts with alternative pre-determinate modes that operate efficiently for a particular task or situation.
Chunjie et al., 2007	Network control <b>Informatics</b>	propose the use of a genetic algorithm to reconfigure dynamically the layout of a networked control systems. The reconfiguration mechanism takes a representation of the architecture (i.e. routing path and encoding scheme) and changes its representation according to the objective of minimizing the network-time delay.
Patouni et al., 2007	Adaptive Systems <b>Communications</b>	introduces advanced mechanisms for autonomic decision making and self-configuration in the communication network. The reconfiguration enable optimized attribution of resources with game theory techniques in order to select the new decision procedure within the system architecture.
Youssef et ElMaraghy, 2007	Reconfigurable systems <b>Manufacturing</b>	propose an approach that selects the configurations according to the current situation while taking into consideration the smoothness of the anticipated reconfiguration process from one configuration to the next expected configuration. This approach uses hybrid metaheuristics in order to optimize the reconfiguration process.
Yu, Zhang, and Klemm, 2007	Execution systems <b>Manufacturing</b>	addressed the reconfiguration of multi agent system aiming to realize an optimal operation of a manufacturing system while simultaneously considering alternative configurations and structures for the system. The reconfiguration is achieved by changing and reorganizing the resource logical address of the agent according to a predefined pattern from a blackboard.
Gumzej et al., 2009	Embedded Systems <b>Computer Science</b>	propose a reconfiguration pattern for distributed systems. The reconfiguration is performed centralized

		by a manager and changes pre-defined protocols contained in the components according to the system needs.
Rooker et al, 2009	Distributed system <b>Manufacturing</b>	propose an approach for a component-based development reconfiguration based on the standard IEC 61499. It takes advantage of the function blocks of this standard in order to have alternative function block ready to be implemented when it is necessary.
Holvoet, Weyns and Valckenaers (2009)	Assembly lines <b>Manufacturing</b>	proposes Delegate MAS as an integrated coordination technique for self-organising coordination and control applications for systems. The reconfiguration in this approach consist in creating exploratory ants in order to foresee possible behaviour of the holonic architecture.
Hsieh et al, 2010	Holonic system <b>Manufacturing</b>	addresses formal models in a reconfiguration mechanism of a holonic architecture. During a perturbation, a method evaluate the impact and, locating the affected components, it activate new protocols for this holons to change the architecture of the system.
Renna, 2010	Production control <b>Manufacturing</b>	proposes a policy to manage capacity exchange among manufacturing processes. A centralized agent calculates the capacity needed in the manufacturing resources in order to fulfil an order demand. The flexibility in the capacity of the resources allow this reconfiguration, but a minimization of cost is bounding the capacity changes.
Valente et al., 2012	Production control <b>Manufacturing</b>	propose a scheduling approach for supporting the dynamic execution of control tasks in distributed control systems while enriching the control nominal behaviour with data gathered from the shop-floor. The reconfiguration mechanism considers a schedule of control task and have inference over the addition or removal over this component, as well as considering the completion of the assigned tasks.
Raileanu et al. (2012)	Assembly lines <b>Manufacturing</b>	propose a framework for implementing a switch of operating states between three different production strategies, each with its own planning objective and perturbation avoidance capabilities.
Eddahech et al.,2013	Embedded systems <b>Computer Science</b>	propose an architecture based on hierarchical multilevel neural network to model workload variation of each task. The neural network replicates the task of each components and the reconfiguration mechanism choose according the needs of the embedded system.
Oh et al., 2013	Logistics <b>Supply Chain</b>	proposes a method of reconfiguring the supply network of an enterprise to cope with its flexibility strategies. The approach has different general configuration strategies that are pre-defined and they are chosen according to the market conditions and a evaluation of the strategy with fuzzy logic.
Ostroumov et al., 2013	Computer architecture <b>Computer Science</b>	aim to incorporate reallocation and reconfiguration procedures into agents hierarchy, which are organized in a three-level structure of a agent management system. The reconfiguration transform the sequences of instructions into a coarse-grained array configuration for evaluating the new mode.
Novas et al. (2013)	Assembly lines <b>Manufacturing</b>	propose a framework that integrates a centralized and a distributed system to minimize the gap between the

		scheduling and the execution operation via delegate MAS and PROSA. The reconfiguration mechanism is that each agent has an exploratory solution that is seen in advance in order to choose the best course of action.
Brusafferi et al., 2014	Cyber physical systems <b>Manufacturing</b>	propose creating a virtual counterpart of the physical system that that enables a reconfigurable and performance optimizing self-similar and hierarchical automation architecture. The virtual part is checking possible routing in a flexible conveyor that improve the scheduled instructions.
Dennis et al., 2014	Middleware approaches <b>Computer Science</b>	propose a hybrid agent architectures for controlling autonomous systems ensuring that dynamic reconfiguration of agents is viable. The reconfiguration mechanism is changing the sequencing of applications of agents according pre-defined states (Order of process).
Sedighizadeh et al., 2014	Power distribution network <b>Energy</b>	propose an approach to minimize the power losses in distribution energy network through network reconfiguration and capacitor placement simultaneously. The reconfiguration mechanism changes a representation of the control architecture solution via particle swarm optimization techniques.
Da Silva et al. 2014	Distributed Control <b>Manufacturing</b>	propose a model based on combination of HMAS and SOA with a fault-tolerant control mechanism. The reconfiguration features are based on the negotiation between holons to change the holarchy according to the priority of the holon task.
Streit et al., 2014	Automation systems <b>Manufacturing</b>	proposes an automatic redeployment of control software during runtime avoiding high costs for redundant hardware. The reconfiguration is in the operations of the components as a relocation of activities is performed once is detected a failure.
Pach et al. (2014)	Assembly lines <b>Manufacturing</b>	proposes a system able to dynamically and partially switch between a hierarchical predictive architecture and a heterarchical reactive architecture, in a perturbation event. Each entity changes the autonomy from receiving instructions to act by its own according to a reactive technique.
Zambrano Rey et al. (2014)	Assembly lines <b>Manufacturing</b>	propose an strategy for controlling online the task schedules with a distributed arrival-time technique and with the objective to support a genetic algorithm offline schedule. The reconfiguration is that the components calculate autonomously an improvement of the schedule changing the behaviour of the decisional entity.
Zielinsly et al, 2015	Control architecture <b>Robotics</b>	propose a to split the control system between the robot embedded control and the cloud computational resources. The reconfiguration in this case is that a dynamical agent with different characteristics takes control of core agent and fix the problem when a perturbation occurs.
Indriago et al, 2015	Hybrid system <b>Manufacturing</b>	aim to evaluate the performance of a predictive-reactive control using a holonic approach applied to a switch arrival system. The reconfiguration featured is guided by switching the server buffer causing a change in the operation schedule.

Liu et al, 2015	Design process <b>Mechatronics</b>	proposes a design for the conception of mechanism or products with the capability of having different modes. The reconfiguration principle lies in including a specific functionality of each operating mode and consider the transition from one state to the other. The premise is that at the end is an user (Human or another component) that chooses the operating state.
Oliveira and Barbosa, 2015	Self adative software <b>Computer Science</b>	build on a framework for formal verification of architectural requirements, which will leverage analysis and adaptation prediction. The approach sets predefined strategies in order to be steer them by connectors between the software components.
Michalos et al. 2015	Production robotics <b>Manufacturing</b>	investigates the flexibility aspects of production systems that use highly interactive and autonomous mobile robotic units for the manufacturing assembly line. The reconfiguration featured in this approach is the replacement of a perturbed robot with a redundant one.
Barbosa et al, (2015)	Assembly lines <b>Manufacturing</b>	proposes an evolution to the ADACOR holonic control architecture inspired by biological and evolutionary theories, where it is included a self-organization mechanism that changes behavioural and structural when it is needed. The reconfiguration is with specific reasoning modules that define how to evolve in structure and behaviour according to a pheromone technique.
Borangui et al (2015)	Assembly lines <b>Manufacturing</b>	propose a control mechanism with a bidirectional switching of an operating mode in order to ensure as long as possible both global optimisation and agility to changes in batch orders under perturbed events. The reconfiguration approach predefine three different states that have its own strategy and objectives to respond to perturbed events.
Jimenez et al. (2015)	Assembly lines <b>Manufacturing</b>	propose a control system with a governance mechanism that, via a configuration array, change the strategy of control within the system architecture. The reconfiguration featured is according the evaluation of the alternative configurations during a perturbation.
Dias-Ferreira et al., 2016	Bio-inspired systems <b>Manufacturing</b>	create a bio-inspired reference architecture focused on highly dynamic environments and the capacity of adhere the bio-inspired principles for self-organizing purposes. The reconfiguration presents is the inclusion of different patterns or templates that are expect to arrange the control architecture in certain scenarios.
Gerostathopoulos et al., 2016	Cyber physical systems <b>Computer Science</b>	propose a model targeting that addresses self-adaptability and supports dependability by providing traceability between system requirements, distinct situations in the environment, and predefined configurations of system architecture. The reconfiguration mechanism changes between different pre-defined configuration that are uses according to different scenarios.
Gorecky et al., 2016	Industrie 4.0 <b>Manufacturing</b>	propose a system architecture of modular production for discrete flow shop. The topology of the system permits adding a component when ever is needed for modularity purposes. This characteristic is change the

		control architecture and change the planning and scheduling of the system.
Leitao and Barbosa, 2016	Holonic system <b>Manufacturing</b>	extend the ADACOR reference architecture by describing the switching mechanism that supports this dynamic balance and global and local forces for the self-organization model. The reconfiguration is a change of cooperation between the holons according to the dissemination of the pheromones over the network.
Vasconcelos et al., 2016	Data streaming <b>Computer Science</b>	propose date a non-disruptive reconfiguration approach for distributed data stream systems that support components and intermittent connections in mobile communications. The reconfiguration feature is the changes of the components variables in order to changes the behaviour and communication protocols, while assuring that the new alternative follow some network rules
Xia et al., 2016	Maintainance <b>Manufacturing</b>	propose an approach of a dynamic maintenance strategy for those reconfigurable structures in manufacturing operations. The reconfiguration is related to the scheduling of maintenance to the machines that change according to possible disruptions online.

# Appendix E

## Modules of the reconfiguration mechanism

### a. Pseudocode for the reconfiguration triggering module

```

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  $t_0, t_1, \dots, t_{max}$ 
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  $\alpha_1$  and  $\beta_1$  for minimum and/or maximum tolerance for  $f(t, X)$ 
10:  Set thresholds  $\alpha_2$  and  $\beta_2$  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  $\alpha_2 \leq g(t, X, S) \leq \beta_2$ 
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

```

### b. Pseudocode for the reconfiguration technique module

```

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  $U_h$ 

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

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

```

### c. Pseudocode for the reconfiguration technique module

```

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  $\Delta X$ 

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  $\Delta X$  by the synchronization technique  $W$ 

Step 5: Implement operating mode
12:   Set new operating mode  $U_{solution}$  in system and consider the synchronization changes  $\Delta X$ 

```

## **Dynamic and hybrid architecture for the optimal reconfiguration of control systems: Application to manufacturing control**

Discrete-event control systems have the opportunity to resolve significant challenges of modern society. In particular, these represent a fundamental solution to manage and control the new technological advances in compliance to the increased consciousness of sustainable development. The parameterization, configuration and decision-making of these control systems are critical aspects that impact the performance and productivity required. Dynamic control architecture approaches, such as reconfigurable control systems, have been proposed for modelling such systems. However, such approaches have failed to address the recovery of the reconfiguration process as these focus on the continuity of execution rather than on the optimisation of the reconfiguration. This dissertation proposes a reference architecture for a reconfigurable control system, named Pollux, designed to manage and adjust optimally and in real time the architecture of a control system, either to guide operational execution or to respond to a system perturbation. Considering a proposed framework of an optimal configuration of control architectures based on shared governance, this proposed approach aims to orchestrate a flexible and customizable decisional entity, a representation that characterizes the unique configuration and control solution of the control architecture, and a three-module reconfiguration mechanism that integrates the optimality-based principles into the reconfiguration process, to ensure a recovery of global performance and/or minimise the degradation caused by perturbations. Our approach is applied in the manufacturing domain and is validated in a simulation and a real flexible manufacturing system cell located at the University of Valenciennes, France. The validation conducted in three experimental scenarios verified the benefits of our approach and encourage us to continue research in this direction.

**Keywords:** Reconfigurable system, Control systems, Optimization, Reactivity, Dynamic reconfiguration, Multi agent systems, MAS, Operating mode, Governance, FMS, Discrete-event control systems.

## **Architecture dynamique et hybride pour la reconfiguration optimale des systèmes de contrôle : Application au contrôle de fabrication**

Les systèmes de contrôle des événements discrets ont la possibilité de résoudre les défis importants de la société moderne. En particulier, cela représente une solution fondamentale pour gérer et contrôler les nouvelles avancées technologiques en conformité avec la requête du développement durable. Le paramétrage, la configuration et la prise de décision de ces systèmes de contrôle sont des aspects critiques qui influent sur les performances et la productivité. Les approches d'architecture de contrôle dynamique, telles que les systèmes de contrôle reconfigurables, ont été proposées pour la modélisation de ces systèmes. Cependant, ils n'ont pas réussi à optimiser le processus de reconfiguration car celles-ci se concentrent sur la continuité de l'exécution plutôt que sur l'optimisation de la reconfiguration. Cette dissertation propose une architecture de référence pour un système de contrôle reconfigurable, nommé Pollux, conçu pour gérer et ajuster de manière optimale et en temps réel l'architecture d'un système de contrôle, soit pour guider l'exécution opérationnelle ou répondre à une perturbation du système. En considérant une proposition d'une configuration optimale des architectures de contrôle basées sur la gouvernance partagée, cette approche proposée un système de contrôle reconfigurable composé d'une entité décisionnelle flexible et personnalisable, d'une représentation qui caractérise la configuration unique et la solution de contrôle de l'architecture de contrôle et d'un mécanisme de reconfiguration à trois modules qui intègre les principes basés sur l'optimalité dans la reconfiguration. Notre approche est appliquée dans le domaine de la fabrication et est validée dans une simulation et une cellule réelle de fabrication située à l'Université de Valenciennes, en France. La validation effectuée dans trois scénarios expérimentaux a permis de vérifier les avantages de notre approche et de nous encourager à continuer la recherche.

**Mots-clés:** système reconfigurable, systèmes de contrôle, optimisation, réactivité, reconfiguration dynamique, systèmes multi-agents, MAS, mode de fonctionnement, gouvernance, FMS, systèmes de contrôle des événements discrets.

---