



HAL
open science

Application du paradigme holonique aux systèmes de contrôle hybrides

Carlos Indriago

► **To cite this version:**

Carlos Indriago. Application du paradigme holonique aux systèmes de contrôle hybrides. Automatique. Université de Nantes; Universidad de los Andes (Mérida, Venezuela), 2015. Français. NNT : 2015NANT2077 . tel-02481529

HAL Id: tel-02481529

<https://hal.science/tel-02481529>

Submitted on 17 Feb 2020

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.

Thèse de Doctorat

Carlos Indriago

*Mémoire présenté en vue de l'obtention du
grade de Docteur de l'Université de Nantes
Docteur de la Universidad de los Andes
Sous le label de l'Université Nantes Angers Le Mans*

Discipline :
Spécialité : Automatique et Informatique Appliquée
Laboratoire : Institut de Recherche en Communications et Cybernétique de Nantes IRCCyN

En vue d'une soutenance le 27 Novembre 2015

École doctorale : Sciences et technologies de l'information et mathématiques
Thèse N°

Application du paradigme holonique aux systèmes de contrôle hybrides

JURY

Rapporteurs : **Sallez, Yves**, Maître de conférences HDR à l'ENSIAME, Université de Valenciennes et du Hainaut-Cambrésis, LAMIH
Thomas, André, Professeur à l'ENSTIB d'Epinal, Université de Lorraine, CRAN

Examineur: **Pujo, Patrick**, Maître de conférences HDR à Polytech' Marseille, Université d'Aix-Marseille, CRET-LOG
Cardillo, Juan, Professeur à l'Université des Andes, Mérida, Venezuela, LaSDAI

Directeur de thèse : **Castagna, Pierre**, Professeur à l'IUT de Nantes, Université de Nantes, IRCCyN

Co-directeur de thèse : **Chacòn, Edgar**, Professeur à l'Université des Andes, Mérida, Venezuela, LaSDAI

Co-encadrant : **Cardin, Olivier**, Maître de conférences à l'IUT de Nantes, Université de Nantes, IRCCyN

Membre invité : **Rakoto, Naly**, Maître Assistant à l'Ecole des Mines de Nantes, IRCCyN

Remerciements

Mes remerciements et toute ma gratitude tout d'abord à Dieu et à ma femme Carolina, soutien fondamental dans les moments les plus difficiles.

Mes parents, qui sont des êtres exceptionnels avec lesquels je vais toujours être. A ma famille, qui trouvent toujours un mot d'encouragement lorsqu'il est nécessaire.

Mes tuteurs Pierre Castagna et Edgar Chacón, mes guides au doctorat. Et un remerciement très spécial à Olivier Cardin, sa patience et son intégrité a rendu possible ce voyage.

A Naly Rakoto et Odile Morineau, pour leur précieuse contribution à la mise en œuvre de ce travail.

A mes amis à Nantes, Claude, Jean-Jacques, Annie, David et Francisco. Ils ont rendu mon séjour à Nantes beaucoup plus facile.

Résumé

De nombreux travaux s'intéressent à l'interaction entre des process de fabrication continue et des algorithmes de planification discrets, ce qui a créé une classe de systèmes hybrides appelées « Hybrid Control Systems » (HCS). Ces systèmes peuvent être décomposés en trois niveaux distincts : les niveaux contrôleur, atelier et interface. Les HCS sont traditionnellement modélisés par des systèmes à commutation représentant le système complet plutôt que des ressources atomiques. De ce fait, les process de reconfiguration sont complexes car il est nécessaire de considérer le système dans sa globalité, rendant les HCS peu flexibles et plus sensibles aux incertitudes. Ce besoin de flexibilité a donc poussé plusieurs équipes à travailler sur l'application du paradigme holonique aux HCS.

L'objectif de cette thèse est de démontrer la possibilité d'appliquer pratiquement directement l'une des architectures de référence holonique des systèmes à événements discrets aux HCS. Pour ce faire, nous proposons une nouvelle architecture appelée Holonic Hybrid Control Model (H²CM), basée sur l'architecture PROSA, et son application sur les HCS.

Deux cas d'études sont présentés pour vérifier le fonctionnement correct du modèle proposé ; le premier est un exemple typique de la littérature des systèmes hybrides appelé Système à Commutations d'Arrivée, tandis que le second est inspiré d'un process industriel de génération d'électricité, spécifiquement un Process à Cycles Combinés. Ces deux cas d'études ont été simulés et un ensemble d'expérimentations a été mené afin d'évaluer les caractéristiques de flexibilité, autonomie, coopération, performance et stabilité du système de contrôle obtenu.

Mots clés: Hybrid Control Systems, architectures de référence holonique, Holonic Hybrid Control Model, Système à Commutations d'Arrivée, Process à Cycles Combinés.

Table des matières

Table des Figures	ix
Resumen en español.....	1
Avant-Propos	3
Introduction.....	5
Chapitre 1.....	9
1 Introduction aux systèmes de contrôle hybrides et systèmes holoniques.....	9
1.1.4 Autonomie des systèmes hybrides	18
1.2 Evolution du mode de pilotage vers plus de flexibilité.....	19
1.3 Systèmes de production holoniques	23
1.3.1 Définitions et caractéristiques.....	23
1.3.2 Architectures de référence Holoniques.....	25
1.3.3 Exemple d'application des HMS pour le contrôle d'un SED.....	29
1.4 Revue du chapitre.....	31
Chapitre 2.....	35
2 Application du paradigme holonique aux systèmes hybrides.....	35
2.1 Travaux antérieurs.....	35
2.1.1 Approches multi-agent.....	35
2.1.2 Approches holoniques.....	39
2.1.3 Discussion.....	48
2.2 Holonic hybrid control model	50
2.2.1 Structure fractale de H ² CM.....	51
2.2.2 Définition du holon ordre.....	52
2.2.3 Définition du holon produit	52
2.2.4 Définition du holon ressource	54
2.3 Planification et supervision de la production	55
2.4 Supervision du système.....	56
2.5 Revue du chapitre.....	58
Chapitre 3.....	59
3 Cas d'étude de systèmes de commutation d'arrivée.....	59
3.1 Description du système de commutation d'arrivée.....	59

Table des matières

3.1.1	Notations utilisées.....	60
3.2	Modélisation du système de commutation d'arrivée.....	61
3.2.1	Programmation JAVA	62
3.2.2	Fonction objectif du pilotage	62
3.3	Modèle d'ordonnancement optimal du nombre de changements de serveur	63
3.4	Holonic Hybrid Control model sur des systèmes de commutation d'arrivée.	64
3.5	Flexibilité de la commande	65
3.6	Optimalité de la commande.....	68
3.6.1	Définition des instances	68
3.6.2	Expérimentations	70
3.6.3	Résultats et discussion	71
3.7	Stabilité de la commande	75
3.7.1	Définition des expérimentations	75
3.7.2	Résultats et discussion	76
3.8	Revue du chapitre.....	78
Chapitre 4.....		79
4	Cas d'étude de centrale thermique à cycle combiné.....	79
4.1	Description du process de cycle combiné	79
4.2	Modèles du cas d'étude	81
4.2.1	Modèle de turbine à gaz	81
4.2.2	Modèle de turbine à vapeur.....	84
4.2.3	Modélisation Java des turbines	86
4.3	Application de H ² CM sur un Process à cycle combiné.....	87
4.4	Définition des expérimentations.....	90
4.5	Résultats	92
4.5.1	Planning	92
4.5.2	Flexibilité	94
4.5.3	Performance	94
4.6	Revue du chapitre.....	98
Conclusion et perspectives.....		99
Références.....		103

Table des Figures

Figure 1. Isomorphisme entre le système et son modèle.	10
Figure 2. Classification des systèmes étudiés.	11
Figure 3. Exemple de système de maintien en température utilisant un automate hybride.	14
Figure 4. Exemple de réseau de Petri hybride.	15
Figure 5. Modèle hybride d'Antsaklis-Stiver-Lemmon.....	16
Figure 6. (a) Système de réservoir (b) Modèle avec le système de contrôle hybride	17
Figure 7. Automate du contrôleur.	17
Figure 8. Système de n réservoirs et un serveur	18
Figure 9. Paires mode d'opération fi et contrôle gi	19
Figure 10. Structure d'un produit orienté information.....	21
Figure 11. Architecture de pilotage contrôlé par le produit (Gouyon et al., 2007)	22
Figure 12. Holons vs. Eléments traditionnels	24
Figure 13. Blocs de base d'un HMS et leurs relations.....	26
Figure 14. Méta-modèle PROSA	27
Figure 15. Agrégation de holons Ressources.....	28
Figure 16 Spécialisation de holons Ressources.	29
Figure 17 Structure du système de traitement de surface.	30
Figure 18 Implémentation au sein d'un outil industriel.....	32
Figure 19. (a) Système de contrôle hybride, (b) Représentation holonique pendant le process de planification, (c) Représentation holonique pendant le process de supervision.	33
Figure 20. Modèle d'un agent multiservice	36
Figure 21. Description des agents	37
Figure 22. Structure multi-agent	39
Figure 23. Framework de contrôle.....	41
Figure 24. Algorithme de type H^∞ dans le Framework à recherche d'objectifs.	41
Figure 25. Relations entre les trois éléments d'une PU	44
Figure 26. Unité de Production.....	45
Figure 27. Fonctionnalités des éléments basiques de contrôle du process DRPC.....	47
Figure 28. Distributed reconfiguration process	48
Figure 29. Du HCS (a) au H ² CM (b).	50
Figure 30. Diagramme de blocs élémentaires de H ² CM.....	51
Figure 31. Exemple de liste de services pour fabriquer un produit.	53
Figure 32. Définition de service.....	53
Figure 33. Modèle de communication des holons Ressource.....	55
Figure 34. Structure d'un holon Ressource.....	55
Figure 35. Comparaison entre l'architecture de référence PROSA (gauche) et H ² CM (droite).....	56
Figure 36. Diagramme de séquence des mécanismes de négociation.	57
Figure 37. Expression des états du superviseur d'une ressource composée.	58

Table des Figures

Figure 38. Système de commutations d'arrivée avec $n=3$.	60
Figure 39.(a) Région représentée par toutes les trajectoires possibles de $x_i(t)$, (b) avec bornes inférieures non-nulles, (c) avec bornes inférieures et supérieures.	62
Figure 40. Diagramme de classes du système des réservoirs.	63
Figure 41. Modélisation holonique du système	65
Figure 42. Algorithme de définition du planning de réservations de la position du serveur	66
Figure 43 (a) Comportement du niveau des réservoirs, deux réservoirs vident au même temps. (b) Réservation des réservoirs, (c) Planification et positionnement du serveur.	67
Figure 44. Algorithme de supervision de la position du serveur	67
Figure 45. Evolution de la proportion d'instances inadaptées au contrôle classique de (Chase et al., 1993) en fonction de l'horizon de simulation	68
Figure 46. Evolution des niveaux de 3 puis 10 réservoirs en fonctions du nombre d'itérations	69
Figure 47. Position du serveur en fonction du nombre d'itérations pour 3 puis 10 réservoirs	69
Figure 48. Scénario 1, 2000 itérations, timelapse de 34	72
Figure 49. Scénario 2, 2000 itérations, timelapse de 16	72
Figure 50. Scénario 3, 3000 itérations, timelapse de 37	72
Figure 51. Scénario 4, 2000 itérations, timelapse de 37	73
Figure 52. Influence du timelapse initial sur les résultats de simulation du scénario 2 sur 2000 pas	73
Figure 53. Analyse de stabilité du contrôle holonique.	77
Figure 54. La turbine à Gaz.	80
Figure 55. La turbine à vapeur.	80
Figure 56. Configuration basique d'un CCP.	81
Figure 57. Le cycle de Brayton.	82
Figure 58. Modèle HGDT de (Rowen, 1983)	82
Figure 59. Modèle HDGT issu de Matlab – Modèle de référence.	85
Figure 60. Comparaison de la simulation du modèle Java et du modèle Matlab de la turbine.	86
Figure 61. Modèle de Turbine à vapeur et d'HRSG simplifié.	86
Figure 62. Diagramme de classe du programme Java de CCP.	87
Figure 63. Modélisation holonique du CCP 4x2.	88
Figure 64. Diagramme de séquence de planification du CCP.	89
Figure 65. Algorithme « Compute best solution ».	91
Figure 66. Planification et exécution de la production.	97

Resumen en español

Numeroso son los trabajos que se interesan en la interacción entre los procesos de fabricación continua y los algoritmos de planificación discretos, lo que creó una clase de sistemas híbridos llamados "sistemas de control híbridos" (HCS por sus siglas en inglés). Estos sistemas pueden ser divididos en tres niveles distintos: el nivel del controlador, el nivel del proceso y el nivel de la interfaz. HCS son tradicionalmente modelados por sistemas de conmutación que representan la totalidad del sistema en lugar de sus recursos de manera atómicos. Por lo tanto, el proceso de reconfiguración es complejo, ya que es necesario considerar el sistema como un todo, haciendo que el HCS sea poco flexible y más sensible a las incertidumbres. Esta necesidad de flexibilidad ha llevado a varios equipos a trabajar en la aplicación de un paradigma holónico sobre los HCS.

El objetivo de esta tesis es demostrar la posibilidad de aplicar casi directamente una arquitectura de referencia holónicas para sistemas de eventos discretos sobre los HCS. Para ello, se propone una nueva arquitectura llamada Modelo de control híbrido holónico (H²CM por sus siglas en inglés), basado en la arquitectura PROSA y su aplicación a el HCS.

Se presentan dos casos de estudio para verificar el correcto funcionamiento del modelo propuesto; el primero es un ejemplo típico de la literatura de los sistemas híbridos denominados sistema de conmutación de llegada, mientras que el segundo está inspirado en un proceso industrial de la generación eléctrica, en concreto una planta de ciclos combinados. Se simularon estos ejemplos y se llevaron a cabo una serie de experimentos para evaluar las características de flexibilidad, autonomía, cooperación, rendimiento y estabilidad del sistema de control.

Palabras Claves: Sistemas Híbridos de Control, Arquitecturas de referencia holónicas, Modelo de control híbrido holónico, Sistema de conmutación de llegada, Planta de ciclo combinado.

Avant-Propos

Ce document présente des travaux de thèse de doctorat réalisés dans le cadre d'une cotutelle internationale entre l'Université de Nantes, France, et l'Universidad de Los Andes, Merida, Venezuela.

Cette thèse s'est déroulée au sein de l'équipe Analyse et Commande des Systèmes à Evènements Discrets de L'IRCCyN de Novembre 2011 à Novembre 2015.

Cette cotutelle a été rendue possible par le financement partiel du projet ECOS-Nord V11M01.

Introduction

Les systèmes de production sont traditionnellement divisés en trois classes principales, i.e. systèmes continus, à événements discrets et hybrides. La classe des process discrets s'intéresse aux systèmes pour lesquels des produits individuels sont fabriqués en utilisant des opérations variées associées librement (telles que des opérations d'usinage CN ou de fabrication additive par exemple) puis assemblés au sein de lignes d'assemblage de produits semi-finis pour créer un produit final. Le contrôle de ces systèmes est classiquement optimisé via des techniques d'ordonnancement, capable de déterminer le meilleur ordre des produits à fabriquer afin de maximiser ou minimiser un critère de performance donné, tel que le makespan ou le retard moyen. L'autonomie est réalisée par l'existence de superviseurs permettant d'assurer l'obtention du comportement souhaité du système. Cependant, cette approche totalement prédictive est progressivement remplacée par des approches prédictive-réactives, capable de réagir aux perturbations apparaissant lors de l'application de l'ordonnancement obtenu sur l'atelier (Aytug et al., 2005). (Mehta and Uzsoy, 1998) ont classé ces approches en quatre catégories principales : approches complètement prédictives, approches prédictive-réactives, ordonnancements robustes et ordonnancements basé sur les connaissances.

Une autre voie explorée est de redéfinir l'architecture de contrôle de manière à donner de l'autonomie d'action et de décision à des entités spécifiques. Ce concept est classiquement repris sous la dénomination de paradigme holonique, à l'origine de la création au milieu des années 1990 des systèmes de production holoniques (Holonics Manufacturing Systems – HMS). Les architectures de contrôles holoniques (Van Brussel et al., 1998)(Leitão and Restivo, 2006)(Chirn and McFarlane, 2000) sont des architectures flexibles permettant une reconfiguration en ligne des process. Une architecture de production holonique a pour objectif de permettre une (auto-) configuration simplifiée, une extension et modification du système aisées, et permet de générer plus de flexibilité et de plus grands espaces de décisions offerts aux entités.

La classe des process continus met en jeu des flux continus de produits, tels que de l'eau ou des produits chimiques, au travers de réseaux interconnectés de tuyauterie. Leur comportement est généralement décrit à l'aide d'un ensemble d'équations différentielles, décrivant l'évolution des sorties du système considérant la valeur des entrées et des différents paramètres pouvant être adressés. Le contrôle optimal de ce type de système nécessite généralement l'inversion du système d'équations, i.e. la détermination du meilleur jeu de paramètres permettant d'obtenir les sorties désirées connaissant les entrées. Un système continu est autonome s'il est capable de se réguler seul, i.e. il existe une fonction qui permettant l'ajustement du comportement du système pour obtenir un objectif défini.

Les systèmes hybrides sont des systèmes dynamiques où le comportement général est déterminé par l'interaction entre des dynamiques continues et des dynamiques discrètes. L'intérêt d'étudier cette classe est qu'elle représente un grand nombre de systèmes, tant dans les industries pharmaceutiques, alimentaires ou hydrocarbures. Beaucoup de familles de systèmes hybrides existent. Parmi elles, ce document se focalisera sur la famille des Systèmes de Contrôle Hybrides (Hybrid Control Systems – HCS) (Antsaklis, 1998). Ce type de systèmes est composé d'un outil de production linéaire continu dans le temps et à temps invariant décrit par équations différentielles, qui inclut des variables dont la valeur évolue linéairement avec le temps. Cet outil est lié à un contrôleur linéaire discret dans le temps et à temps invariant décrit par des équations linéaires aux différences, qui inclut des variables dont la valeur évolue de manière discrète avec le temps. L'autonomie d'un HCS implique l'autonomie de chaque élément ayant un comportement discret et une autonomie dans le contrôle des sauts de comportement. Selon (Koutsoukos et al., 2000), les HCS consistent en trois niveaux distincts. Tout d'abord, le contrôleur est une machine séquentielle, vu comme un système à évènements discrets, qui reçoit, manipule et émet des évènements représentés par des symboles. Le process est un système à états continus destiné à être contrôlé in fine par le contrôleur à états discrets, qui reçoit, manipule et émet des signaux représentés par des variables réelles typiquement continues par morceaux. Le contrôleur et le process communiquent via une interface, qui traduit les signaux du process en symboles pour que le contrôleur puisse les utiliser (via un générateur) et vice versa (via un actionneur). Les HCS sont conventionnellement modélisés par des systèmes à commutations concernant le système complet. De ce fait, le processus de reconfiguration d'un tel système est complexe car le modèle considère le système comme un tout indissociable ce qui complexifie les calculs lors d'une modification du modèle. De ce fait, les HCS sont réputés pour être peu flexibles et sensibles aux aléas, et de nombreux problèmes traités dans la thématique traitent de flexibilité, autonomie, stabilité, détection de défaillances ou réactivité.

Avec l'objectif de trouver des solutions à ces problématiques, l'utilisation d'architectures de contrôle intrinsèquement flexibles et autonomes a été étudiée à plusieurs reprises. Parmi ces travaux, plusieurs équipes ont étudié en détail l'application du paradigme holoniques aux HCS (McFarlane, 1995)(Chacón et al., 2002)(Chokshi and McFarlane, 2008). Leur objectif a été de développer un système de contrôle permettant de se conformer aux exigences et spécificités des systèmes hybrides en se basant sur des applications industrielles réelles. De ce fait, le résultat obtenu est cohérent et fonctionnel, mais les holons créés le sont sur la base des fonctions classiques exécutées dans le contrôle conventionnel. Bien qu'efficace, cette approche tend à manquer de généralité et ne permet pas d'envisager une réutilisation à large échelle de ces approches ni une utilisation du grand nombre de résultats issus de la communauté des évènements discrets.

L'objectif de cette thèse est de démontrer la possibilité d'appliquer quasiment à l'identique l'architecture de référence PROSA (Van Brussel et al., 1998) au contexte des HCS. Ce document est divisé en quatre chapitres :

Introduction

- Le chapitre 1 est une introduction aux systèmes de contrôle hybrides et systèmes holoniques.
- Le chapitre 2 passe en revue les quelques travaux traitant des approches hétérarchiques sur les systèmes continus, puis une explication du modèle proposé appelé H²CM.
- L'objectif du chapitre 3 est d'appliquer le modèle proposé sur un cas d'étude classique de la littérature et de comparer ses performances par rapport à une solution optimale calculée offline grâce à un algorithme d'ordonnancement. Les notions de stabilité et de performance du système de contrôle proposé sont donc évaluées au travers de l'application à un cas d'étude simple.
- Le chapitre 4 concerne l'application à un cas d'étude de centrale thermique à cycle combiné. Ce cas d'étude a été choisi autour d'un process industriel de génération d'électricité, ce qui permet de montrer la généricité de l'approche et son déploiement sur un système de beaucoup plus grande dimension.

L'avancement de ces travaux a été régulièrement publié lors de congrès internationaux (Indriago et al., 2015a, 2014a, 2014b) et ont fait l'objet d'un chapitre d'ouvrage (Indriago et al., 2015b). Ils ont de plus fait l'objet de trois soumissions d'articles dans des revues internationales (Computers in Industry, Concurrent Engineering Research and Applications et Journal Européen des Systèmes Automatisés), celle soumise à Computers in Industry étant en révision mineure au moment de l'écriture de ce document.

Chapitre 1

1 Introduction aux systèmes de contrôle hybrides et systèmes holoniques.

Cette thèse concerne la modélisation d'un type des systèmes hybrides appelé les systèmes de contrôle hybrides en utilisant des systèmes holoniques. Un système de contrôle hybride est un système composé d'un process de production continu (ou semi-continu) piloté avec des événements discrets. Ces systèmes seront modélisés pour étudier les caractéristiques comme la flexibilité et la réactivité.

Les deuxièmes systèmes à étudier sont les systèmes holoniques, lesquels sont des systèmes de pilotage qui possèdent des caractéristiques de flexibilité, caractéristiques que nous souhaitons appliquer sur les systèmes de contrôle hybrides.

Ainsi, dans ce chapitre est présentée une introduction aux deux types de systèmes qui seront étudiés, d'abord les systèmes de contrôle hybrides puis les systèmes holoniques.

1.1 Systèmes de contrôle hybrides

Quelques définitions sont données afin de mieux comprendre les systèmes de contrôle hybrides. Nous nous intéressons aux systèmes qui existent physiquement. Un système est un objet où plusieurs éléments interagissent entre eux, et comme résultat de cette interaction un certain comportement est produit (Villa Ramirez, 2004).

D'après les idées de (Von Bertalanffy, 1973), un système est un ensemble de variables qui maintiennent une relation fonctionnelle dans le temps, où l'état actuel d'une variable donnée dépend de ses propres valeurs passées aussi bien que des autres variables. De là, la définition de systèmes dynamiques, où la sortie dépend des valeurs passées de l'entrée, prend corps. Les équations différentielles ou les équations aux différences sont utilisées pour décrire le comportement du système dynamique. Nous nous intéressons seulement

aux systèmes avec lesquels nous pouvons interagir, en d'autres termes, des systèmes avec des entrées et des sorties.

Afin de travailler avec le système, nous développons un modèle, qui est un autre système ayant un comportement équivalent a minima pour les variables auxquelles nous nous intéressons. Selon (Cassandras and Lafortune, 2008), on peut considérer comme modèle un élément qui duplique simplement le comportement du système lui-même. Plus précisément, nous devons développer des méthodes d'abstraction pour décrire le comportement.

Selon (Ashby, 1956), « les deux systèmes,..., sont ainsi reliés qu'un homomorphisme de l'un est isomorphe d'un homomorphisme de l'autre ». Comme il existe un isomorphisme entre le système et son modèle, quand nous parlons des systèmes dynamiques, nous parlons des modèles dynamiques, comme le montre la Figure 1.

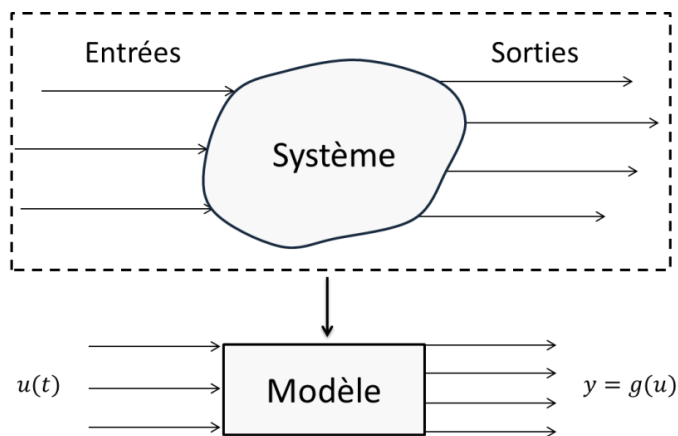


Figure 1. Isomorphisme entre le système et son modèle.

Les définitions suivantes sont issues de (Cassandras et Lafortune, 2008), et décrivent les caractéristiques des systèmes qui seront étudiés. Des systèmes à temps invariant sont des systèmes qui ne varient pas avec le temps. Cette propriété, également appelée stationnarité, implique que nous pouvons appliquer une entrée spécifique à un système et espérer qu'il réponde toujours de la même manière.

La définition de l'état nous permet de décrire les états utilisés qui nous concerne dans les systèmes étudiés, donc, de manière générale, l'état d'un système à un instant t doit décrire son comportement à cet instant d'une manière mesurable. Dans notre cas, les systèmes étudiés peuvent avoir deux types d'états : les systèmes à états continus où les variables d'états peuvent prendre généralement des valeurs réelles ou complexes et les systèmes à états discrets où les variables d'états appartiennent à un ensemble discret (e.g. les entiers non-négatifs). Dans ce cas, une trajectoire typique est une fonction constante par morceaux, puisque les variables d'états peuvent sauter d'un état discret à un autre. D'une part, les systèmes continus sont habituellement modélisés dans le cadre standard des équations différentielles et leur comportement est appelé comportement continu. D'autre part, les systèmes à événements discrets sont modélisés avec des automates à états finis, ou des cadres plus récents comme les réseaux de Petri ou l'algèbre max-plus, et son

comportement s'appelle le comportement séquentiel ou comportement à événements discrets. Naturellement, il existe des situations où un modèle hybride est approprié, c'est-à-dire que certaines variables d'état sont continues, d'autres sont discrètes.

Ainsi, il y a des systèmes orientés par des événements et des systèmes orientés par le temps. Dans les systèmes orientés par le temps, l'état change quand le temps évolue. Dans les systèmes orientés par les événements, seulement l'occurrence d'événements discrets va forcer les transitions d'état. Entre les occurrences d'événements, l'état reste inchangé. Nous souhaitons travailler avec les systèmes pilotés par événements.

Un système est dit stochastique si au moins une de ses variables de sortie est une variable aléatoire. Sinon, le système est dit déterministe. En général, l'état d'un système dynamique stochastique définit un processus aléatoire, dont le comportement ne peut être décrit que d'une manière probabiliste. Pour un système déterministe avec une entrée donnée, l'état peut être évalué à partir de l'instant initial.

Enfin, un système à temps continu est un système pour lequel toutes les variables d'entrées, d'états et de sorties sont définies pour toutes les valeurs possibles du temps. Pour les systèmes à temps discrets, seulement une ou plusieurs variables sont définies à un temps discret donné, généralement résultant d'un processus d'échantillonnage.

Après avoir décrit les définitions de base, le système qui sera utilisé peut être classifié comme le montre la Figure 2. Nous travaillerons avec des systèmes dynamiques, de temps invariants, non linéaires, d'états continus, pilotés par événements, déterministes et de temps discret ou continu. La famille de systèmes que nous allons étudier sont les systèmes de contrôle hybrides, en particulier ceux pour lesquels l'état continu est piloté par des événements.

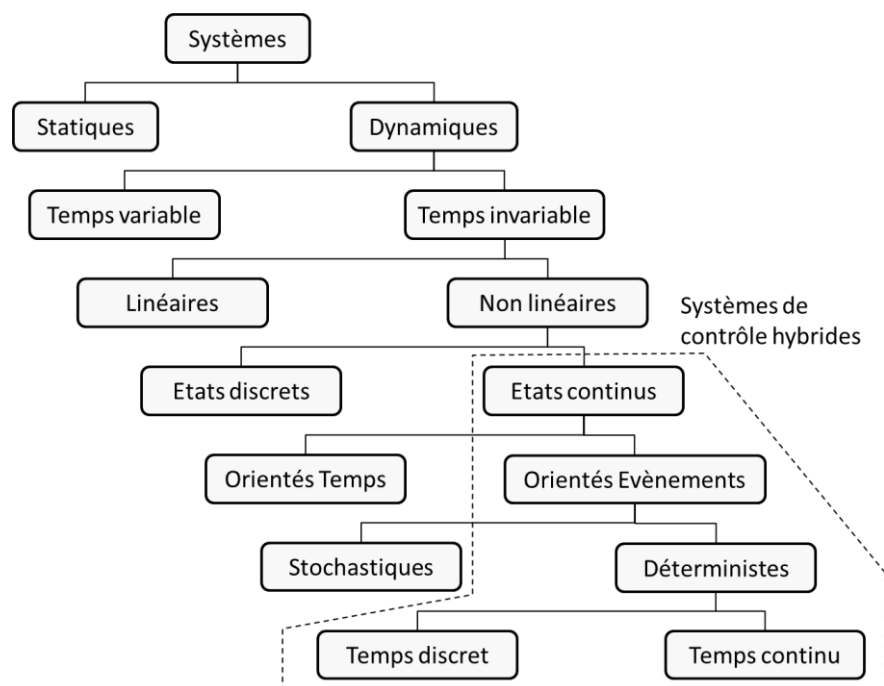


Figure 2. Classification des systèmes étudiés.

Selon (Antsaklis, 1998), les systèmes hybrides sont des systèmes dynamiques dont le comportement est déterminé par des dynamiques continues et discrètes qui interagissent entre elles. Selon (Van Der Schaft et al., 2000), les systèmes hybrides sont un mélange de dynamiques continues et d'évènements discrets en temps réel. Ces dynamiques continues et discrètes non seulement co-existent mais aussi interagissent et des changements se produisent à la fois en réponse à un évènement instantané discret, mais également en réponse à des dynamiques décrites par des équations différentielles ou équations aux différences temporelles.

Selon (Branicky, 1995), les systèmes de contrôle hybrides sont des systèmes qui incluent des dynamiques à la fois continues et discrètes, ainsi que des systèmes de contrôle (de commande) continus et discrets. Les dynamiques continues sont modélisées par des champs de vecteurs contrôlés ou des équations aux différences. La nature hybride provient de la dépendance à quelques phénomènes discrets, correspondant à des états discrets, des dynamiques discrètes ou des commandes (contrôles) discrètes.

Une seconde définition de système de contrôle hybride est donnée par (Antsaklis 1998). Selon lui, il existe plusieurs exemples de systèmes hybrides, mais dans le domaine de l'Automatique, un exemple bien connu est celui où le système continu est décrit par des équations différentielles linéaires et commandé par un système linéaire à temps discret.

1.1.1 Modélisation de systèmes hybrides

Après avoir défini les systèmes hybrides, la prochaine étape est de trouver un modèle approprié pour le système. La modélisation est définie comme l'acte de reproduire le comportement d'un système, il se peut donc que certains comportements soient négligés afin de réduire la complexité du modèle. Les outils généralement utilisés pour modéliser les systèmes hybrides sont introduits par les descriptions suivantes.

Automates Hybrides

Une prolongation normale de la représentation de modèle des automates définis pour les systèmes à événements discrets (Alur et al., 1993) (Nicollin et al., 1993) vers les systèmes hybrides est d'assigner à chaque nœud de l'automate une équation dynamique régissant le comportement du vecteur hybride d'état, et d'employer les transitions entre les nœuds pour représenter les commutations et les sauts.

Un automate hybride $H = (Var, Graph, Init, Inv, Flow, Jump, Event)$ se compose de sept éléments (Henzinger, 2000) :

- *Var* : Un ensemble fini $X = \{x_1, x_2, \dots, x_n\}$ de variables de nombre réels. Le nombre n est appelé la dimension de H . $\dot{X} = \{x_1, x_2, \dots, x_n\}$ représente la première dérivée pendant le changement continu, et $X' = \{x_1, x_2, \dots, x_n\}$ représente les valeurs après le changement discret.
- *Graph* : Un multi graphe fini (V, E) . les nœuds V sont appelés commandes. Les transitions en E sont appelées les commutations commandées.

- *Init, Inv et Flow* sont respectivement les conditions initiales, l'invariant et le flux, qui associent à chaque mode commandé $v \in V$ trois prédicats.
- *Jump*, conditions de saut. Une fonction d'étiquette de transitions qui associe à chaque commutation commandée $e \in E$ un prédicat. Chaque saut $jump(e)$ est un prédicat pour lequel ses variables libres appartiennent à $X \cup X'$.
- *Event*. Un ensemble fini Σ d'événements, et une fonction d'étiquette de transition $event: E \rightarrow \Sigma$ qui associe à chaque commutation commandée un événement.

Un exemple simple est montré par le système de contrôle de température d'une pièce (Alur et al., 1994). La température d'une pièce est contrôlée par un thermostat qui mesure continuellement la température et allume/éteint le chauffage. La température est donnée par des équations différentielles. Quand le chauffage est éteint, la température, désignée par la variable x , décroît suivant la fonction exponentielle :

$x(t) = \theta e^{-Kt}$, où t est le temps, θ est la température initiale et K est une constante déterminée par les caractéristiques physiques de la pièce.

Quand le chauffage est allumé, la température évolue suivant l'équation :

$x(t) = \theta e^{-Kt} + h(1 - e^{-Kt})$, où h est une constante qui dépend de la puissance du chauffage.

L'objectif est de maintenir la température entre m et M degrés en allumant ou éteignant le chauffage selon la Figure 3. L'automate hybride décrivant ce système est composé des éléments suivants:

- *Var* : $X = \{x\}$.
- *Graph* : $(V = \{on, off\}, E = \{(on, off), (off, on)\})$.
- *Init, Inv et Flow*.
 - $init(on) = \{(x^+ = x^-)\}; init(off) = \{(x^+ = x^-)\}$
 - $inv(on) = x > 0; inv(off) = x > 0$
 - $flow(on) = \{\dot{x} = K(h - x)\}; flow(off) = \{\dot{x} = -Kx\}$
- *Jump*, condition de saut. $Jump(on, off) = x \geq M, Jump(off, on) = x \leq m$
- *Event*. $\Sigma = \{m, M\} event: \{(on, off) \rightarrow M; (off, on) \rightarrow m\}$.

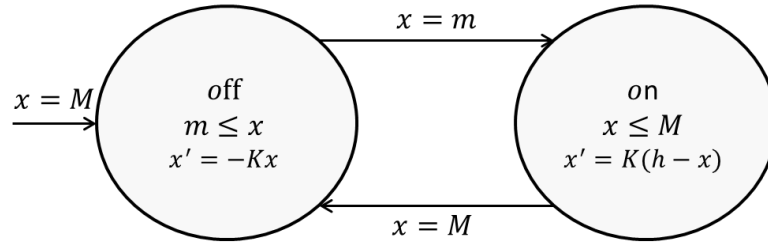


Figure 3. Exemple de système de maintien en température utilisant un automate hybride.

Réseaux de Petri Hybrides

Les réseaux de Petri sont un formalisme introduit par Carl Adam Petri pour modéliser des systèmes avec simultanéité et sont définis en tant que graphes bipartites orientés (Petri, 1966). Quelques prolongements pour modéliser les dynamiques continues ont été proposés dans (David and Alla, 2010). Plusieurs méthodologies pour représenter les systèmes hybrides ont été proposées, quelques exemples sont les Réseaux de Petri Hybrides (Alla and David, 1998), Réseaux de Petri Flux (Trivedi and Kulkarni, 1993) et Réseaux de Petri différentiels (Demongodin and Koussoulas, 1998). Dans cette sous-section, nous présentons le formalisme développé pour les réseaux de Petri différentiels ou réseaux de Petri différentiels temporisés qui permettent de modéliser des variables continues avec des valeurs négatives.

Un réseau de Petri différentiel est défini par $B = [R, f, Mo, time]$, où :

- R est un réseau de Petri défini comme $R = [P, T, Pre, Post]$ avec,
 - P : un ensemble fini de places avec $|P| = n$,
 - T : un ensemble fini de transitions avec $|T| = m$,
 - $P \cap T = \emptyset$ et $P \cup T \neq \emptyset$,
 - $Pre(P_i, T_j)$ est une fonction qui associe des arcs depuis d'une place vers une transition,
 - $Post(P_i, T_j)$ est une fonction qui associe des arcs depuis une transition vers une place,
- $f: P \cup T \rightarrow \{D, DF\}$, appelée « fonction différentielle », s'associe à chaque nœud si c'est un nœud discret ou différentiel.
- Mo est le marquage initial.
- $Time$ est une fonction qui associe un nombre rationnel à chaque transition qui peut évoluer dans le temps, et également un retard pour la transition différentielle (représentant le pas d'intégration).

La Figure 4 montre un exemple de système de mise en bouteille issu de (David and Alla, 2010). Un fût contenant 57 litres de vin est mis en bouteilles de 0.75 litre. Ceci est modélisé par la transition T_2 : cette transition est validée s'il existe au moins 0.75 jetons dans la place P_4 et un jeton dans la place P_1 . La place P_3 indique que les bouteilles doivent être remplies une par une et non simultanément.

Le réseau de Petri différentiel représentant l'exemple a les éléments suivants,

- $R = [P, T, Pre, Post]$
 - $P = \{P_1, P_2, P_3, P_4\}$
 - $T = \{T_1, T_2, T_3\}$
 - $Pre = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0.75 & 0 \end{bmatrix}$
 - $Post = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
- $f = \{P_4 = DF, P_1 = P_2 = P_3 = D, T_1 = T_2 = T_3 = D\}$
- $M(0) = [4 \ 0 \ 1 \ 57]^T$
- $Time = \{\}$

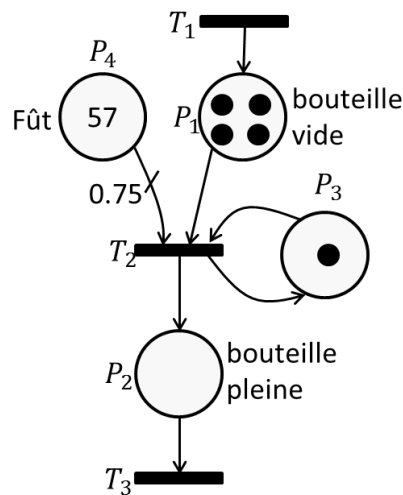


Figure 4. Exemple de réseau de Petri hybride.

Il existe d'autres outils pour modéliser les systèmes hybrides qui ne seront pas détaillés dans ce travail, comme la théorie géométrique de systèmes hybrides (Branicky, 1995) (Polderman and Willems, 2000) (Sontag, 1998), les Bond graphs (Paynter, 1961) (Mosterman, 1997), et les systèmes MLD (Bemporad and Morari, 1999) (Villa Ramirez,

2004). Finalement, nous avons concentré ce travail sur les systèmes de contrôle hybrides modélisés par le modèle hybride d'Antsaklis-Stilver-Lemmon, qui utilise les automates comme présenté dans la section suivante.

1.1.2 Modèle hybride d'Antsaklis-Stilver-Lemmon

Suivant le modèle de (Antsaklis et al. 1998), un système hybride comporte trois niveaux distincts: le process, le contrôleur et l'interface, comme le montre la figure 5.

Le process (plant) est un système continu décrit par des équations différentielles ou des équations aux différences et constitue le système à commander. Le process reçoit, manipule et produit en sortie des signaux représentés par des variables réelles. Le contrôleur est un système à événements discrets, comme par exemple un automate à états finis. Le contrôleur reçoit et manipule et produit en sortie des événements représentés par des symboles qui sont typiquement continus ou continus par morceaux. Le contrôleur et le process communiquent via l'interface. L'interface convertit les sorties du process en symboles pour le contrôleur. Et ce dernier convertit les symboles de sortie du contrôleur en signaux d'entrée pour le process.

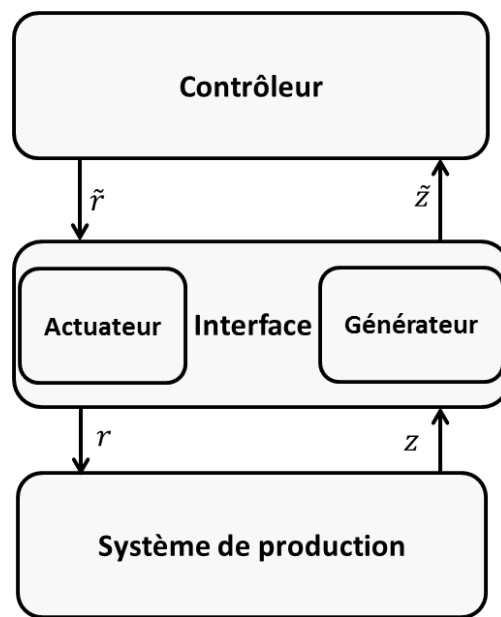


Figure 5. Modèle hybride d'Antsaklis-Stilver-Lemmon

1.1.3 Exemple de systèmes hybrides

Ceci est un exemple pour illustrer comment un système de contrôle hybride peut être modélisé. Le système consiste en un réservoir tampon qui se vide à travers une vanne de sortie, tandis que la vanne d'entrée est contrôlée par un système à événements discrets, comme le montre la figure 6(a). Le contrôleur permet au réservoir de se vider jusqu'à un niveau minimum, et ouvre la vanne d'entrée pour le remplissage. Quand le réservoir a atteint un niveau maximum, la vanne d'entrée est fermée. Le niveau du liquide dans le réservoir est régi suivant l'équation différentielle suivante :

$$\dot{x} = r - x^{1/2}$$

Où x est le niveau du liquide et r est le débit entrant. L'interface partitionne l'espace d'état en trois régions comme suit :

$$\alpha(x) = \begin{cases} \tilde{z}_1 & \text{si } x > \max \\ \tilde{z}_2 & \text{si } \min < x < \max \\ \tilde{z}_3 & \text{si } x < \min \end{cases}$$

Ainsi quand le niveau augmente et atteint \max , le symbole \tilde{z}_1 est généré. Et quand le niveau diminue et atteint \min , le symbole \tilde{z}_3 est généré. L'interface génère ainsi deux entrées correspondant aux deux symboles \tilde{r}_1 et \tilde{r}_2 pour le contrôleur comme suit :

$$\gamma(\tilde{r}) = \begin{cases} 1 & \text{si } \tilde{r} = \tilde{r}_1 \\ 0 & \text{si } \tilde{r} = \tilde{r}_2 \end{cases}$$

Comme $r = \gamma(\tilde{r})$, ceci veut dire que la vanne d'entrée sera ouverte si $r = \tilde{r}_1$, et elle sera fermée si $r = \tilde{r}_2$, Figure 6 (b).

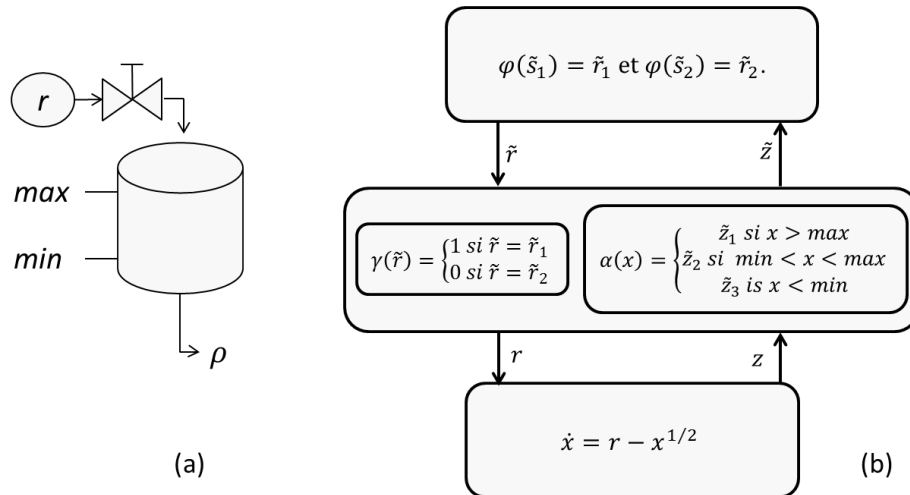


Figure 6. (a) Système de réservoir (b) Modèle avec le système de contrôle hybride

Le contrôleur pour le système de réservoir est un automate à états finis composé de deux états \tilde{s}_1 et \tilde{s}_2 . En partant de l'état \tilde{s}_1 , l'automate va à l'état \tilde{s}_2 si le symbole \tilde{z}_1 est reçu. En partant de l'état \tilde{s}_2 , l'automate va à l'état \tilde{s}_1 s'il reçoit le symbole \tilde{z}_3 . Enfin, il reste dans l'état courant s'il reçoit le symbole \tilde{z}_2 . Par ailleurs, $\varphi(\tilde{s}_1) = \tilde{r}_1$ et $\varphi(\tilde{s}_2) = \tilde{r}_2$ cf. Figure 7..

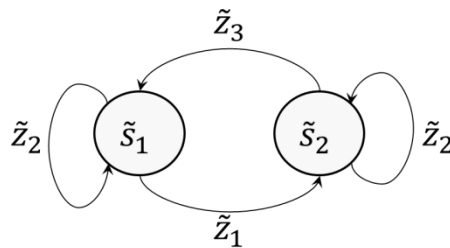


Figure 7. Automate du contrôleur.

Les points forts des systèmes hybrides de modélisation sont étudiés à travers la littérature, (Antsaklis et al., 1993), (Nerode and Kohn, 1993a), (Branicky, 1995), (Van Der Schaft et al., 2000), (Villa Ramirez, 2004). Il existe de multiples outils pour la modélisation et la synthèse de contrôleurs afin de trouver des solutions aux nombreux problèmes des systèmes hybrides. Mais comme les processus étudiés deviennent plus complexes, la tâche de modélisation des processus est elle-même de plus en plus complexe, et une fois modélisé, il devient difficile de modifier le processus si un besoin de reconfiguration apparaît. Pour illustrer cela, nous reprenons le processus de l'exemple ci-dessus, mais au lieu d'avoir un seul réservoir, nous considérons n réservoirs avec toujours un seul serveur, Figure 8. Modéliser ce processus en utilisant le modèle du système de contrôle hybride devient alors une tâche très difficile. En effet, les systèmes de contrôle hybride réalisent la modélisation des processus en tant qu'ensemble indissociable, et pour modifier un élément du processus on doit alors complètement changer le modèle déjà réalisé.

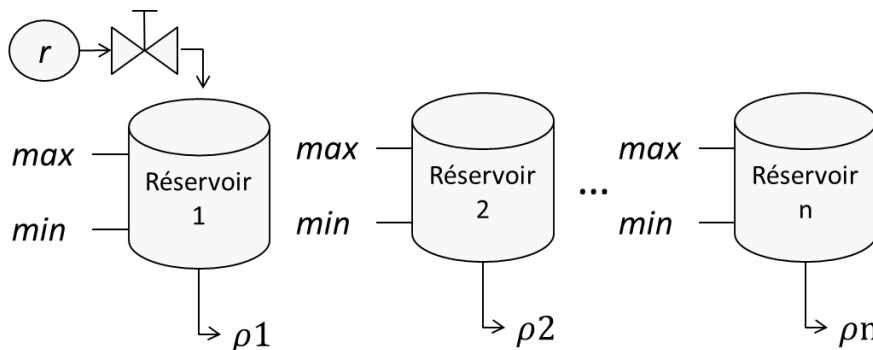
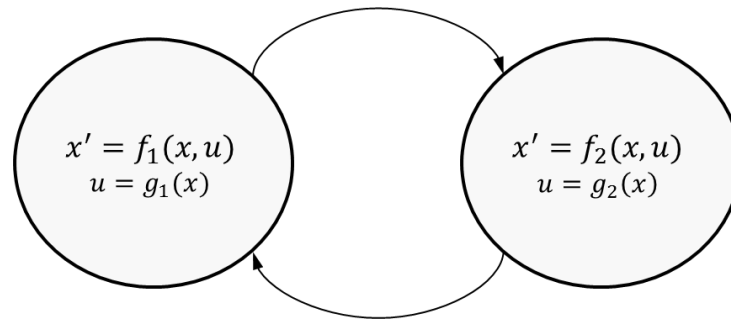


Figure 8. Système de n réservoirs et un serveur

1.1.4 Autonomie des systèmes hybrides

Un système continu est autonome s'il est capable de se réguler seul, i.e. il existe une fonction qui permettant l'ajustement du comportement du système pour obtenir un objectif défini. Pour les systèmes discrets, la définition de l'autonomie est obtenue par l'existence de superviseurs pour assurer le comportement du système désiré. L'autonomie dans un système hybride implique l'autonomie dans chaque état discret du système, et l'autonomie dans le contrôle du saut implicite aux systèmes hybrides (Koutsoukos et al., 2000). Le formalisme de l'autonomie implique l'existence de superviseurs qui permettent d'amener les systèmes aux états de fonctionnement permettant d'être autonomes, possédant un contrôleur régulant la dynamique dans cet état, permettant de rester en l'état, ou de définir un chemin pour atteindre une limite où se produit un saut souhaitée à un nouvel état ou mode d'opération. Le process se compose d'un modèle discret, qui est contrôlé par plusieurs techniques à évènements discrets et des modèles continus qui peuvent être enchaînés par des variables représentatives des dynamiques globales : des vitesses, positions, sauts (qui consistent à modifier les modes d'opération), direction, accélération, changements de débit, etc.

Le suivi des systèmes hybrides implique la présence de paires d'état - contrôleur (contrôle) et d'état - mode d'opération, pour atteindre une limite qui est déterminée par le suivi, Figure 9.

Figure 9. Paires mode d'opération f_i et contrôle g_i .

Le superviseur génère ensuite des lois de contrôle pour chaque état qui permet d'orienter le processus vers un état objectif. Le modèle d' (Antsaklis et al., 1993) peut être ajusté en définissant dans les interfaces la sélection des contrôleurs pour l'état présent qui permet à un certain nombre de sauts d'atteindre le nouvel objectif. (Moor et al., 2002) a proposé l'existence de superviseurs décentralisés pour les systèmes hybrides, où plusieurs superviseurs agissent selon le comportement désiré du process.

Modéliser le process implique la construction de modèles qui peuvent être enchaînés et autonomes au sens de Lipschitz, à savoir chaque mode de fonctionnement correspond à une paire contrôle-mode d'opérations, où les conditions aux limites, lorsqu'elles évoluent, sont détectées et un nouveau mode d'opération est alors associé au contrôle. Le suivi détecte les changements dans le process à l'aide de capteurs spécifiques à chaque mode d'opération, et agit en modifiant la commande, qui doit être compatible avec l'état actuel.

Des entités de contrôle décentralisées, agissant suivant les principes d'autonomie – coopération, et la partie formelle de systèmes autonomes des systèmes hybrides (l'ensemble de la commande et du process) dans un invariant, permettent d'imaginer que l'on soit capable de séquencer les sauts d'un système hybride et de négocier avec les autres entités pour déterminer les régions d'opération compatibles. Chaque entité établirait alors des mécanismes pour assurer les modes d'opérations valides, et des négociations seraient menées pour garantir qu'il n'y a pas de séquence d'évènements discrets qui permette de sortir des régions cohérentes pour atteindre un objectif. La section suivante présente un bref aperçu de ce que ces entités de contrôle décentralisées ont apporté aux systèmes de production manufacturiers en général.

1.2 Evolution du mode de pilotage vers plus de flexibilité

Comme la section précédente l'a démontré, le manque de flexibilité des systèmes de contrôle hybrides provient de la complexité des calculs nécessaires à la synthèse du contrôleur. A cela, plusieurs propositions de solutions peuvent être amenées. Parmi celles-ci, l'opportunité de travailler sur la flexibilité de la partie discrète des HCS permet une décomposition de la partie hybride en éléments de taille plus restreinte, et donc une simplification du contrôle sur cette partie-là. Néanmoins, la complexité se retrouve déportée sur la partie discrète, qui se doit d'avoir un comportement le plus performant possible, à la fois en termes de productivité, flexibilité ou sûreté de fonctionnement. Cette

section propose un bref état de l'art sur les travaux proposant des architectures de contrôle amenant de la flexibilité dans le domaine des systèmes de production discrets.

Depuis moins d'une décennie, l'ensemble de ces concepts se regroupe sous l'égide de la notion de Systèmes Cyber-Physiques (Lee, 2006), et plus particulièrement dans notre cas de Systèmes Cyber-Physiques de Production (Monostori, 2014). Toutefois, cette thématique hérite de beaucoup de travaux connexes réalisés au cours des deux dernières décennies autour de la thématique générale des systèmes de production dits « intelligents ». Cette notion d'intelligence a particulièrement été mise en avant depuis la mise en œuvre de la classification proposée par (Wong et al., 2002), largement acceptée depuis au sein de la communauté. L'idée était de classer les produits en fonction de leurs capacités en deux niveaux d'intelligence, ce qui était très caractéristique des travaux de cette époque, traitant des Systèmes Contrôlés par le Produit (SCP). Un produit est défini comme un objet dual physique et informationnel, doté de capacités de mémorisation, de communication, d'action, de décision qui lui permettent de:

1. Posséder une identification unique;
2. Communiquer avec son environnement;
3. Mémoriser et gérer des informations qui lui sont propres;
4. Disposer d'un langage de dialogue et d'échange de ses informations et états;
5. Participer aux processus de décision durant son évolution;

La notion de décision étant au centre du pilotage par le produit, il nous paraît important de préciser ici le cadre d'une prise de décision. Chaque décision peut être décomposée en 5 caractéristiques (Cardin et al., 2009) :

1. Le Déclencheur de la décision (D): évènement déclenchant la prise de décision.
2. Le Centre de décision (C): entité intelligente du système évaluant la décision.
3. Les Paramètres de la décision (P): Ensemble des données, mesurées, évaluées ou prévues qui influent sur la prise de décision.
4. Le Système Directement impacté par la décision (Sd): Sous-ensemble du système sur lequel la décision a un impact direct.
5. Le Système Influé par la décision (Si): sous-ensemble du système sur lequel la décision a une influence induite.

A partir de ces fonctions de base, (Wong et al., 2002) proposent deux niveaux d'intelligence du produit :

1. A ce niveau, le déclencheur de la décision (D) est l'évènement correspondant à l'arrivée d'un produit en sortie de la machine M1. De même le produit contient tout ou partie des paramètres (P) de la décision. Par contre, le centre de décision est une entité du système de production extérieur au produit. Ce niveau couvre tout autant les produits possédant une identification simple (code-barres, etc.) que les produits captant eux-mêmes de l'information (produits instrumentés) ou encore

les produits possédant une capacité de stockage en lecture/écriture (type étiquette électronique). Ce niveau intègre les fonctions 1 à 3 du produit intelligent défini par Wong et al qui parlent alors de « produit orienté information ».

2. Au niveau 2, le produit est à la fois l'auteur de l'évènement déclencheur et le centre de la décision. C'est lui qui évalue la décision sur la base de paramètres, qu'il peut éventuellement contenir ou recueillir sur le système réel. Ses facultés de communication lui permettent éventuellement de communiquer avec d'autres centres de décisions afin de faire ses choix. Ce niveau intègre les produits capables à la fois d'évaluer la pertinence des solutions possibles et d'interagir avec le système pour l'application de la décision. Ce niveau intègre l'ensemble des fonctions du produit intelligent définies par (Wong et al., 2002).

Comme on peut le voir dans ces définitions, la nature du produit et la relation entre le produit et son environnement sont prépondérantes dans la classification. La mise en place d'un SCP impose d'avoir un produit pour le moins informé voire intelligent, ce qui n'est pas sans poser un problème économique, notamment dans les systèmes haute cadence caractérisés par des flux très importants de produits ayant une faible valeur. Lorsque les encours de production sont relativement faibles, une solution consiste à augmenter le produit en l'associant à un composant qui lui apportera l'intelligence. Ainsi, pour des raisons à la fois économique et d'interopérabilité des moyens, le produit intelligent, dans un système de production, est souvent la combinaison de plusieurs composants. La structure la plus simple étant constituée du produit en cours de fabrication lui-même et d'une base associée au produit, comme par exemple une palette transportant ce produit. C'est l'ensemble produit + base qui constitue le produit orienté information (Figure 10). De façon plus générale, la structure du produit intelligent peut être plus complexe. Considérons un chariot automatisé transportant un bac de produit. L'ensemble forme un produit intelligent de niveau 2. Le bac transporté, lui-même équipé d'une étiquette électronique et d'un système de décision est aussi un produit intelligent de niveau 2. Les produits transportés, équipés de tags RFID sont eux-mêmes un ensemble de produits intelligents de niveau 1.

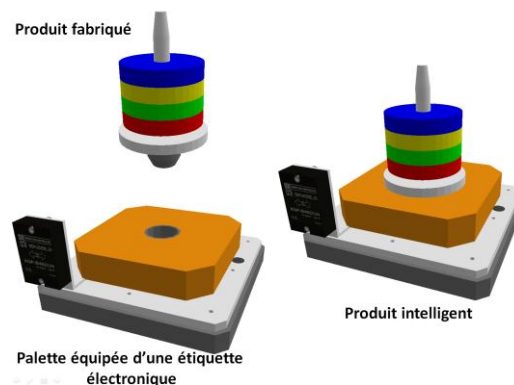


Figure 10. Structure d'un produit orienté information

Le pilotage d'un SCP est donc caractérisé par une architecture distribuée du pilotage, chaque ressource étant contrôlée par un centre de décision (Cardin and Castagna,

2011)(Sallez et al., 2010)(Gamboa Quintanilla et al., 2013). La répartition de l'intelligence permet la simplification du pilotage. Par contre, la prise de décision dans de tels systèmes peut être simpliste, car les paramètres mis à disposition du centre de décision ne concernent généralement qu'un sous-ensemble du système de production (généralement le voisinage du S_d), alors que l'impact de la décision sur le S_i voire sur l'ensemble du système, peut être très significatif. Dans ses travaux concernant la mise en place d'un environnement pour l'évaluation des systèmes contrôlés par le produit, (Pannequin et al., 2009) propose aussi un pilotage distribué, implémenté sur une plateforme multi-agents. Les systèmes multi-agents sont effectivement de bons candidats pour implémenter ces notions d'intelligence distribuée.

Néanmoins, un paradigme plus générique et adapté aux systèmes distribués a été proposé en parallèle afin de guider l'implémentation. Il s'agit du paradigme de systèmes de production holoniques (Holonc Manufacturing Systems – HMS), résultat du projet international HMS (Valckenaers et al., 1998), conjointement avec la notion de systèmes de contrôle hétérarchiques (Aissani et al., 2008). Dans le contexte des systèmes contrôlés par le produit, (Gouyon et al., 2007) proposent une architecture de pilotage basée sur le concept de système de production holonique.

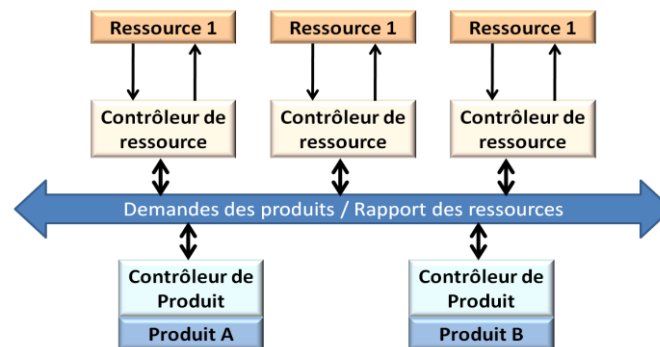


Figure 11. Architecture de pilotage contrôlé par le produit (Gouyon et al., 2007)

Ce système est distribué, basé sur des coopérations entre d'une part des contrôleurs de ressource assurant la bonne exécution des transports, des opérations de production et d'autre part des contrôleurs de produit assurant que les produits subissent la succession des opérations permettant leur fabrication. Beaucoup d'approches ont été utilisées pour étudier le processus de prise de décision, par exemple la stigmergie (Sallez et al., 2009), les champs de potentiel (Berger et al., 2010) ou la négociation via le très répandu Contract-Net Protocol (Borangiu et al., 2014).

Plusieurs architectures de référence holoniques coexistent : PROSA (Van Brussel et al., 1998), ADACOR (Leitão and Restivo, 2006), HCBA (Chirn and McFarlane, 2000), D-MAS ou ORCA-FMS (Pach et al., 2014). Bien que beaucoup d'outils ad hoc aient été développés, le mode d'implémentation générique le plus populaire reste les systèmes multi-agents, donc la plateforme JADE, basée sur Java est le représentant le plus répandu.

1.3 Systèmes de production holoniques

De ce fait, cette section a pour objectif de présenter plus en détail le paradigme holonique appliqué aux systèmes de production. Après quelques définitions générales du paradigme, nous nous concentrerons sur la présentation spécifique de l'architecture PROSA qui sera la base des travaux présentés dans ce document. Enfin, un exemple d'application tiré de la littérature de PROSA à un cas classique de système de production permettra d'illustrer ces concepts.

En préambule, nous rappelons que la notion de système holonique a ses origines dans les travaux de Arthur Koestler, qui a développé les concepts basiques de holons dans son ouvrage « The Ghost in the Machine » (Koestler, 1990). Il a postulé un ensemble de principes sous-jacents pour expliquer la tendance naturelle d'auto-organisation et d'auto-adaptation que l'on peut rencontrer dans les organisations sociales et biologiques. Il a proposé le terme de holon pour décrire les blocs élémentaires de ces structures. Ce terme est la combinaison du mot grec *holos*, signifiant « ensemble », avec le suffixe *-on* signifiant « partie », comme dans proton ou neutron. Ce terme reflète les tendances de holons à agir en tant qu'individus autonomes et coopérants pour former des hiérarchies de sous-systèmes ayant une apparence d'auto-organisation, qu'il a dénommées holarchies.

1.3.1 Définitions et caractéristiques

(Suda, 1989) adapte les concepts de (Koestler, 1990) aux systèmes de production pour la première fois. Ces définitions ont été reprises par le consortium HMS à la fin des années 1990 :

Holon: Bloc élémentaire autonome et coopérant d'un système de production pouvant transformer, transporter, stocker et/ou valider des objets informationnels et physiques. Le holon consiste en une partie traitant l'information et généralement en une autre partie traitant les objets physiques. Un holon peut être une partie d'un autre holon ;

Autonomie: Capacité d'une entité à créer et contrôler l'exécution de ses propres plans et/ou stratégies ;

Coopération: Process pendant lequel un ensemble d'entités développe mutuellement des plans acceptables et les met en œuvre ;

Holarchie: Système de holons qui peuvent coopérer pour atteindre un objectif. La holarchie définit les règles de base pour la coopération des holons, et de fait limite leur autonomie ;

Holonic Manufacturing System (HMS): Holarchie qui intègre l'étendue complète des activités de production, de la conception à la production, en passant par le passage d'ordres de fabrication, afin d'atteindre l'entreprise de production agile.

Un HMS est destiné à permettre la mise en place de systèmes ayant une (auto-) configuration facilitée, des possibilités simplifiées d'extension et de modification et plus de flexibilité en augmentant l'espace de décision aux plus hauts niveaux du contrôle.

L'objectif est de faire bénéficier à la production les bénéfices que les organisations holoniques apportent aux organismes vivants et aux sociétés, i.e. une certaine stabilité face aux perturbations, une adaptabilité et de la flexibilité vis-à-vis du changement et une utilisation efficace des ressources disponibles.

Le concept de HMS concentre naturellement les meilleures caractéristiques des organisations hiérarchiques et hétérarchiques, c'est-à-dire la stabilité de la hiérarchie et la flexibilité dynamique de l'hétérarchie. La Figure 12 présente une comparaison entre un holon et des éléments traditionnels sur une métrique à deux axes en fonction de leur autonomie et de leur habilité à coopérer.

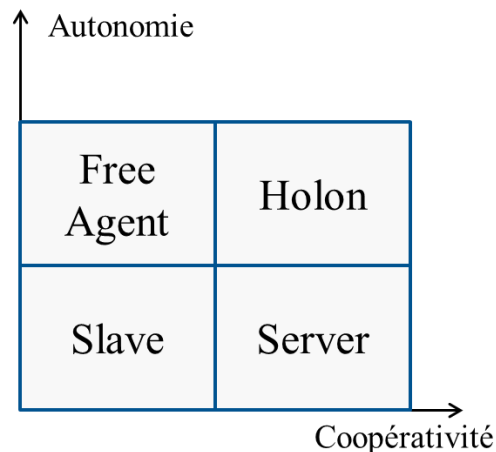


Figure 12. Holons vs. Eléments traditionnels

Les HMS cherchent à améliorer la gestion de facteurs critiques dans la gestion des architectures de contrôle, dont on pourra citer :

- Gestion des perturbations : Fournir de meilleures et de plus rapides capacités de détection et de réponse aux défaillances machines, ordres de fabrication de haute priorité, évolution du rendement des process, erreurs humaines, etc. ;
- Intégration de l'humain dans la boucle de contrôle : Supporter une meilleure intégration de l'intelligence humaine ;
- Disponibilité : Fournir une meilleure fiabilité et faciliter la maintenance même sur des systèmes très grands et complexes ;
- Flexibilité: Supporter une évolution constante dans la conception des produits et des lots de fabrication toujours plus petits.

À partir de cette liste, on peut citer quelques éléments de réponse fournis par les HMS :

1. Aux problèmes de gestion des perturbations et de disponibilité :

- Fournir des systèmes intelligents pour la construction autonome et coopérative des plannings de production, de l'ordonnancement, la détection de défaillances, le diagnostic et leur résolution.

2. Au problème d'intégration de l'humain dans la boucle de contrôle :

- Fournir des interfaces hommes-machines plus intuitives, flexibles et personnalisables ;
- Fournir une « assistance intelligente » pour augmenter l'intelligence humaine et éviter les erreurs humaines.

3. Flexibilité :

- Fournir un contrôle accru de l'humain sur les fonctionnalités et la configuration du système ;
- Fournir des capacités d'auto-reconfiguration ;
- Supporter des changements continus/incrémentaux dans les rôles et relations des éléments du système.

1.3.2 Architectures de référence Holoniques

L'utilisation du paradigme holonique prend tout son intérêt dans ce qu'elle permet d'architecturer de manière plus ou moins générique les systèmes de contrôle distribués. Cette généralité provient de plusieurs travaux de la littérature permettant de dégager, en particulier sur l'application systèmes de production, des architectures de référence, sortes de méta-modèles des futurs systèmes distribués.

Il existe plusieurs architectures de référence holoniques dans la littérature. Parmi elles, on peut citer HoMuCs (Langer, 1999), HCBA (Chirn and McFarlane, 2000), D-MAS (Verstraete et al., 2008), PROSIS (Pujo et al., 2009), ORCA-FMS (Pach et al., 2014) ou encore ADACOR (Leitão and Restivo, 2006) et sa récente évolution (Barbosa et al., 2015). Toutes ces architectures ont pour objectif le contrôle de systèmes de production, et sont très typées vers les systèmes à événements discrets. Dans le cadre de ce document, nous nous intéressons à la déclinaison de ce type d'architectures aux systèmes hybrides. De ce fait, dans cette section, nous nous intéresserons à PROSA (Van Brussel et al., 1998), qui est parmi toute l'architecture la plus générique et la moins typée. Nous pensons ainsi que son application sera facilitée, et que des déclinaisons futures pourront montrer la compatibilité (ou l'incompatibilité) des concepts présentés dans ce document sur la base de PROSA avec les architectures sus-citées.

Architecture holonique *Ordre, Produit, Ressource, Staff* (PROSA)

L'architecture PROSA consiste à la base en trois types de holons: *Ordre*, *Produit* et *Ressource*. Chacun est responsable d'un aspect du contrôle de la production, que ce soit la logistique, la planification technologique ou les capacités des ressources respectivement. Ils sont structurés par des concepts orientés-objets d'agrégation et de spécialisation. Un quatrième type d'holon, *Staff*, peut être ajouté pour assister les holons de base avec une expertise bien précise.

Holons de base

Autant en recherche qu'en industrie, trois problématiques relativement indépendantes existent en production :

1. Les aspects liés aux ressources, comme piloter une machine à vitesse optimale ou maximiser sa capacité ;
2. Les aspects technologiques liés aux produits ou aux process, tels que la détermination des opérations nécessaires pour réaliser un produit ayant une qualité suffisante ;
3. Les aspects logistiques liés aux clients, à la demande et aux dates de livraisons.

À partir de cette analyse, (Van Brussel et al., 1998) a proposé d'identifier les holons présentés sur la Figure 13.

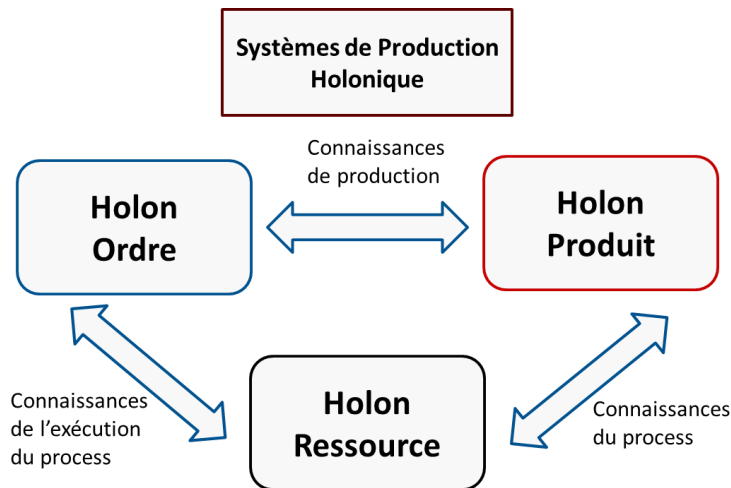


Figure 13. Blocs de base d'un HMS et leurs relations.

Le holon Ressource contient une partie physique, en l'occurrence une ressource du système de production, et une partie capable de traiter de l'information servant au contrôle de la ressource. Il offre des capacités et fonctionnalités de production aux holons environnants. Il contient les méthodes permettant d'allouer les ressources de production, la connaissance et les procédures pour organiser, utiliser et contrôler ces ressources pour mener la production. Un holon ressource peut représenter une abstraction des moyens de production tels que les usines, ateliers, machines, convoyeurs, pipelines, palettes, composants, outils, outillages, stockages, personnels, énergies, zones d'atelier, etc. (Serugendo et al., 2004)

Le holon Produit contient les connaissances sur les process produits permettant d'assurer une correcte réalisation du produit, garantissant également une qualité suffisante. Un holon produit, gère des informations cohérentes et met à jour le cycle de vie du produit, les contraintes utilisateur, la conception, la gamme de réalisation, la nomenclature, les procédures d'assurance qualité, etc. Ainsi, il reconstruit un « modèle du produit » du type de produit, et non un « modèle de l'état du produit » d'une instance physique du produit en cours de production. Le holon produit agit comme un serveur d'information pour les autres holons du HMS.

Un holon Ordre représente une tâche du système de production. Il est responsable de correctement réaliser le travail qui lui a été assigné dans les délais. Il a la gestion des

produits physiques en production, du modèle d'état du produit et tout le traitement des informations logistiques liées à la tâche. Un holon Ordre peut représenter les ordres de fabrication émanant du client, les ordres destinés au stockage, des ordres de fabrication de prototypes, de maintenance ou de réparation de ressources, etc.

Ces trois types de holons échangent des informations concernant le système de production. Les holons Produits et Ressources communiquent des informations sur le process, les holons Produits et les holons Ordres échangent des informations sur la production, et les holons Ressources et les holons Ordres partagent des informations sur l'exécution du process, Figure 14.

- Les informations de process représentent les informations et méthodes sur la manière de réaliser un certain process sur une certaine ressource. Elles informent sur les capacités d'une ressource, les process qu'elle peut réaliser, les paramètres de process pertinents, la qualité pouvant être escomptée, etc.
- Les informations de production représentent les informations et méthodes sur la manière de réaliser un certain produit sur certaines ressources. Elles informent sur les séquences de process possibles à exécuter sur les ressources, les structures de données nécessaires à représenter les sorties des process, etc.
- Les informations d'exécution de process représentent les informations et méthodes sur la progression de l'exécution des process sur les ressources. Elles informent sur la manière de lancer des process sur les ressources, sur le processus de réservations des ressources, la surveillance de la progression de l'exécution, l'interruption des process et ses conséquences, etc.

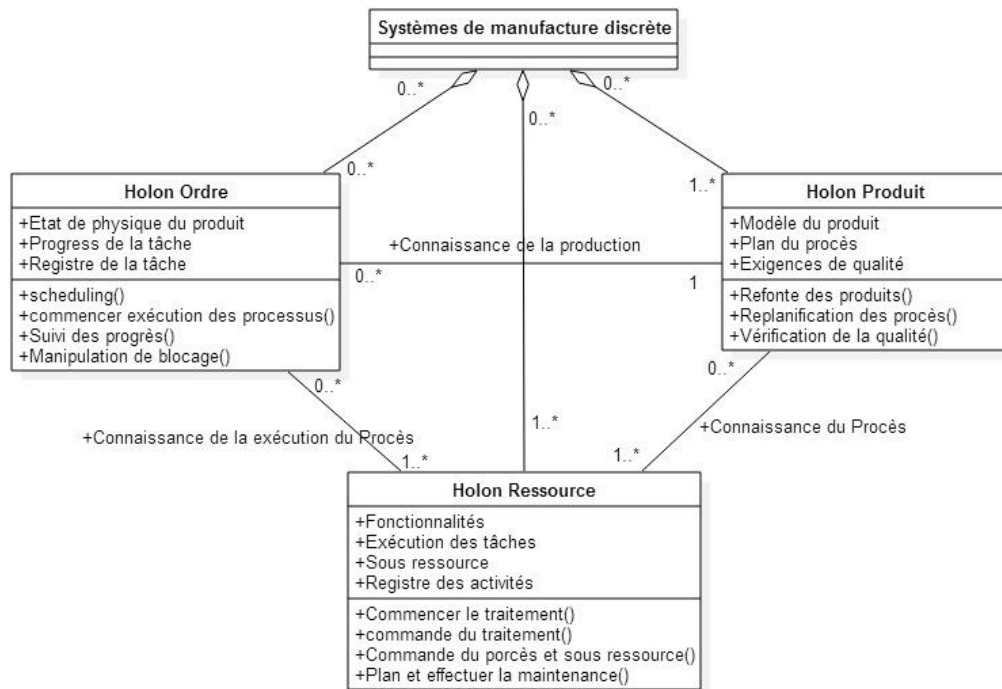


Figure 14. Méta-modèle PROSA

L'architecture résultante a un haut degré d'auto-similarité, ce qui réduit la complexité d'intégration de nouveaux composants du système et permet une configuration ou une reconfiguration plus aisée et rapide. PROSA permet de couvrir à la fois les aspects de contrôle hiérarchiques ou hétérarchiques. Ainsi, l'architecture peut être considérée comme une généralisation de ces deux approches de contrôle. Plus important encore, PROSA introduit des innovations significatives : la structure du système est découplée de l'algorithme de contrôle, les aspects logistiques peuvent être découplés des aspects techniques et PROSA ouvre l'opportunité de réussir à créer des algorithmes de contrôle hybrides plus évolués.

Structurer des holons dans une hiérarchie est une solution appropriée à ce problème de complexité. De ce fait, des holons agrégés sont définis comme un ensemble de holons regroupés et qui forment à leur tour un plus gros holon. Les holons agrégés ne sont pas des ensembles statiques de holons, mais peuvent changer de périmètres dynamiquement selon les besoins du système, Figure 15. La fonction d'agrégation du paradigme holonique est une caractéristique importante dans ce document de par la nature des systèmes étudiés qui sont généralement très grands.

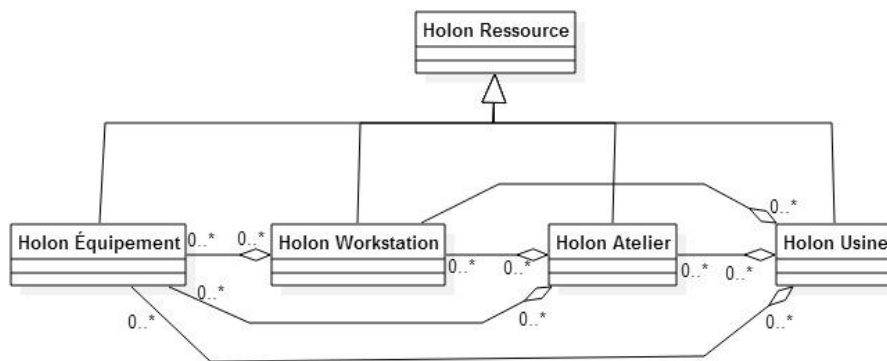


Figure 15. Agrégation de holons Ressources.

La spécialisation sépare les holons en fonction de leurs caractéristiques, à l'intérieur même des holons génériques que sont les Ordres, Produits et Ressources. Pour une architecture spécifique, ces trois holons de base peuvent s'avérer être trop abstraits pour construire efficacement le raisonnement de conception de l'architecture de contrôle. La spécialisation peut alors être utilisée pour différencier les différentes sortes de holons manipulés, Figure 16.

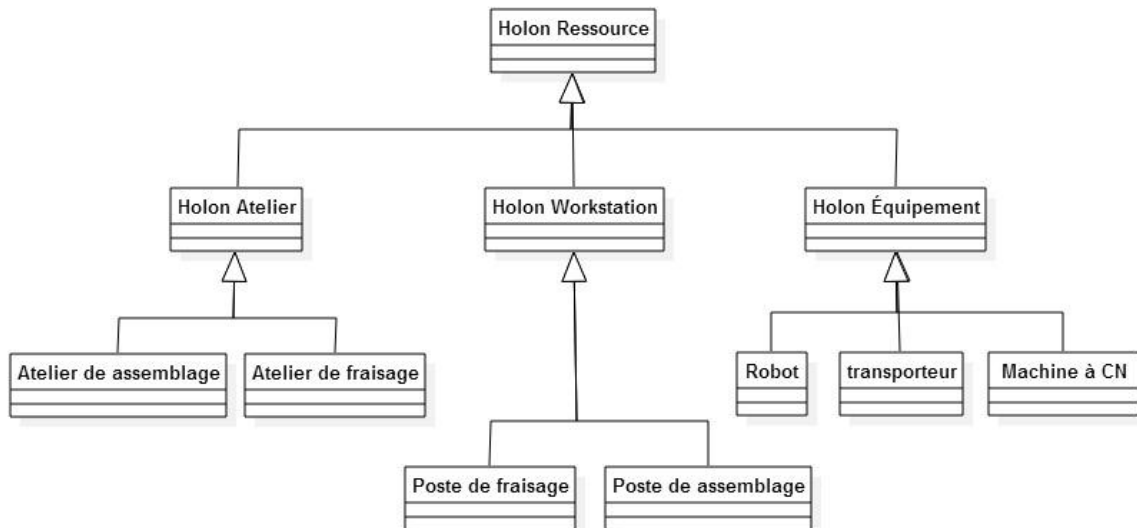


Figure 16 Spécialisation de holons Ressources.

Le Holon Staff

L'architecture offre également la possibilité d'utiliser un holon Staff. Celui-ci permet aux holons de base de mener à bien leurs missions, généralement en les aidant à prendre une décision en leur fournissant un ensemble de données sur le problème en ayant un point de vue différent. Le Staff est en quelque sorte considéré alors comme un expert extérieur donnant un conseil.

Ce concept aide généralement à implémenter des fonctionnalités centralisées dans les architectures de ce type. Cela a deux avantages : d'une part, cela permet de résoudre certains problèmes qui sont trop complexes à résoudre efficacement de manière distribuée, et ensuite cela aide à faire une transition en douceur entre des architectures de contrôle classique généralement centralisées et une architecture holonique où beaucoup d'éléments et fonctions sont décentralisées.

Cependant, l'utilisation du holon Staff n'introduit pas la rigidité qui accompagne la hiérarchie dans le système, puisque la décision finale est laissée aux holons de base. Quand des holons Staff sont ajoutés, les holons de base essaient autant que possible de suivre leurs conseils. Lorsque, à cause de perturbations et de changements intervenant dans le système, les holons Staff ont une performance dégradée, leurs conseils peuvent être complètement ignorés par les holons de base, qui reprennent une complète autonomie de décision. D'un autre côté, lorsqu'aucune perturbation n'intervient, le HMS peut être configuré de telle manière que les holons de base suivent systématiquement les recommandations du Staff (voie hiérarchique). Cette architecture laisse ouverte la possibilité d'implémenter des comportements intermédiaires entre la pure hiérarchie et la pure isoarchie.

1.3.3 Exemple d'application des HMS pour le contrôle d'un SED.

De nombreux développements HMS sont présents dans la littérature, que ce soit au niveau académique (Sallez et al., 2010)(Borangiu et al., 2010) (Pach et al., 2014)

(Dubromelle et al., 2014) ou dans l'industrie (Blanc et al., 2008) (Le Mortellec et al., 2013) (Aissani et al., 2009). Cette section a pour objectif de décliner les concepts présentés dans la section précédente sur un exemple à la frontière entre les deux, étant réalisé dans un cadre académique sur la base d'une problématique industrielle (Chové et al., 2009). Cette application est d'autant plus révélatrice des potentialités amenées par le concept qu'elle a été intégrée dans des outils industriels préexistants pour une application dans l'industrie concernée.

L'application concerne un système de traitement de surface de l'industrie aéronautique. La structure du système est illustrée sur la Figure 17, et a les spécificités suivantes :

- Il y a deux lignes de production (notées Line N°1 et Line N°2) contenant des bacs de traitements chimiques, électrolytiques ou de rinçage;
- Il n'y a qu'un pont desservant chaque ligne;
- Il existe deux stations d'entrées/sorties (notées Input/Output#1 et Input/Output#2);
- Plusieurs produits ayant des gammes différentes peuvent être traités simultanément sur la ligne. Une gamme est définie par une séquence de bacs qui doivent être visités dans l'ordre ;
- Les temps de traitement dans chaque bac sont généralement définis à l'aide d'intervalles $[T_{min} ; T_{max}]$. Si l'une ou l'autre des bornes de l'intervalle n'est pas respectée, les produits concernés sont considérés comme rebutés, ce qui a un énorme impact sur la productivité du système au vu du coût unitaire moyen élevé de chaque pièce ;
- Les ponts sont placés sur des rails et utilisés pour transporter les produits d'un bac à l'autre ;
- Les produits peuvent être transférés de la Line#1 à la Line#2 par un système automatisé dédié localisé en bout de lignes.

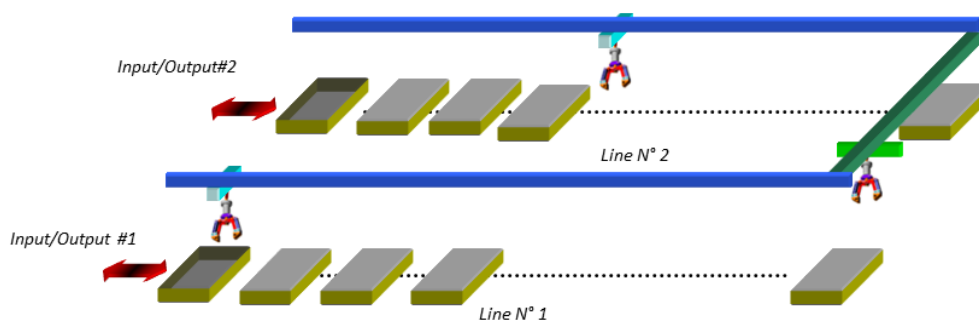


Figure 17 Structure du système de traitement de surface.

Ce cas d'étude est reconnu pour être un problème d'optimisation très complexe du fait des nombreuses contraintes exprimées pouvant être synthétisé selon la classification de (Manier and Bloch, 2003) comme :

RHSP | 2, 1, synchro, (1, 20, Ø), (1, 20, Ø) // ass | 500 / 100, 25, Ø, Ø | C_{max}

Les travaux décrits dans (Chové et al., 2009) ont consisté à proposer une approche hétérarchique, combinant heuristiques réactives et ordonnancements prédictifs en ligne, basée sur le paradigme holonique. Cette approche est décrite en utilisant la terminologie provenant de l'architecture de référence PROSA.

Le holon « Produit » est un holon contenant les descriptions techniques des produits traités sur la ligne. Il contient les connaissances sur le produit et les process, et correspond physiquement à la base de données du Manufacturing Execution System dans lequel il est implémenté.

Le holon « Ordre » est créé lorsqu'un nouvel ordre de fabrication est demandé au système. Il est relié au holon Produit correspondant de manière à déterminer la séquence de bains à visiter. L'objectif de ce holon est d'exécuter la séquence donnée sans violer les contraintes de temps et en assurant une qualité optimale des produits. De ce fait, il gère à la fois les produits physiques à traiter, le modèle d'états des produits et toutes les informations logistiques liées aux opérations.

Les holons « Ressources » sont successivement réservés par les holons Ordre pour réaliser les séquences. Trois éléments constituent la classe des holons Ressources :

1. Les baignoires, où les produits subissent les traitements ;
2. Les positions d'attente, où les produits peuvent être placés de manière à attendre la disponibilité d'un bain ;
3. Les ponts, capables de saisir, déplacer et déposer les produits dans les baignoires.

Leur objectif est d'établir le meilleur planning possible de réservations des ressources de manière à satisfaire les requêtes des holons Ordre tout en respectant leurs propres contraintes (capacité limitée, durées de maintenance, etc.). Toutefois, ce comportement induisant des heuristiques purement réactives ne permettant pas d'assurer à tout instant le respect de la qualité des produits, surtout avec une hausse de la charge du système.

De ce fait, un holon « Staff » a également été défini. Son rôle est d'implémenter une heuristique prédictive destinée à assister la création des plannings. Dans ce cas précis, si le prochain bain qui nécessite d'être réservé à un intervalle de tolérance sur la durée de traitement faible, alors le holon Ressource correspondant au pont concerné demande au holon Staff de réserver une date dans son planning de manière à assurer sa disponibilité lors du départ du produit depuis le bain.

Cette application a été intégrée par un logiciel d'ordonnement classique du commerce, Preactor, qui l'a implémenté à l'aide d'une technologie multi-agents à partir des développements réalisés dans le cadre des travaux de (Chové et al., 2009) (Figure 18).

1.4 Revue du chapitre

L'objectif de ce travail est d'appliquer les systèmes holoniques sur les systèmes hybrides, comme indiqué précédemment à la recherche d'un système de contrôle hybride plus flexible et réactif. La Figure 19 (a) montre la structure classique du système de contrôle

hybride qui sera modifiée en utilisant une approche holonique. Ce changement va s'appliquer en deux étapes : l'étape de planification et celle de la surveillance. La Figure 19 (b) montre comment l'interface et le processus de production sont représentés dans un holon ressource au sein de la structure holonique et comment des signaux envoyés et reçus par le contrôleur pendant le processus de négociation sont interrompus pour trouver un contrôleur en fonction des ressources disponibles. La Figure 19 (c) montre la structure de commande qui est configurée et ressemble à la structure classique du système de commande hybride, à la différence que, si dans le processus de production, un changement se produit, alors le système passe à la structure de négociation holonique pour la reconfiguration du dispositif de contrôle en fonction de ces changements.

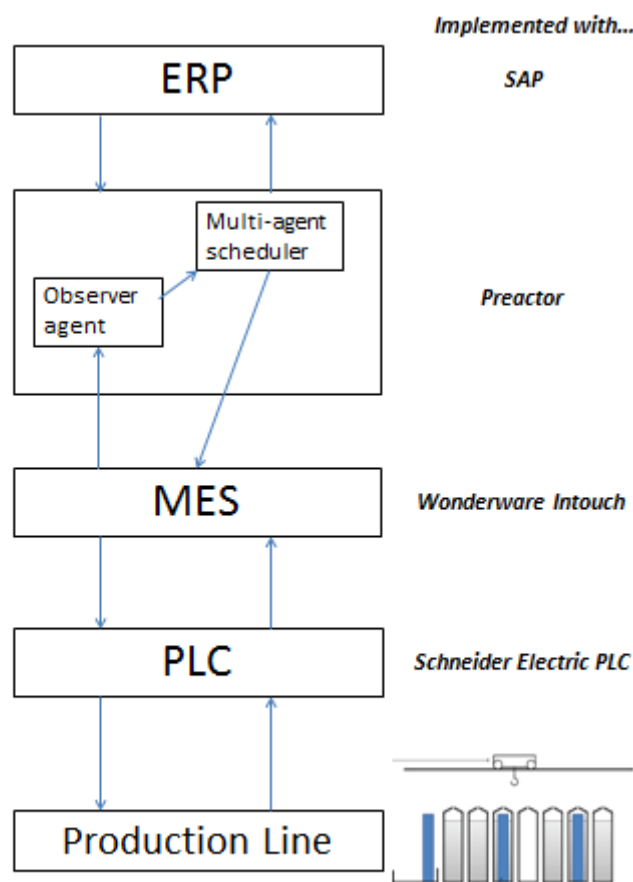


Figure 18 Implémentation au sein d'un outil industriel.

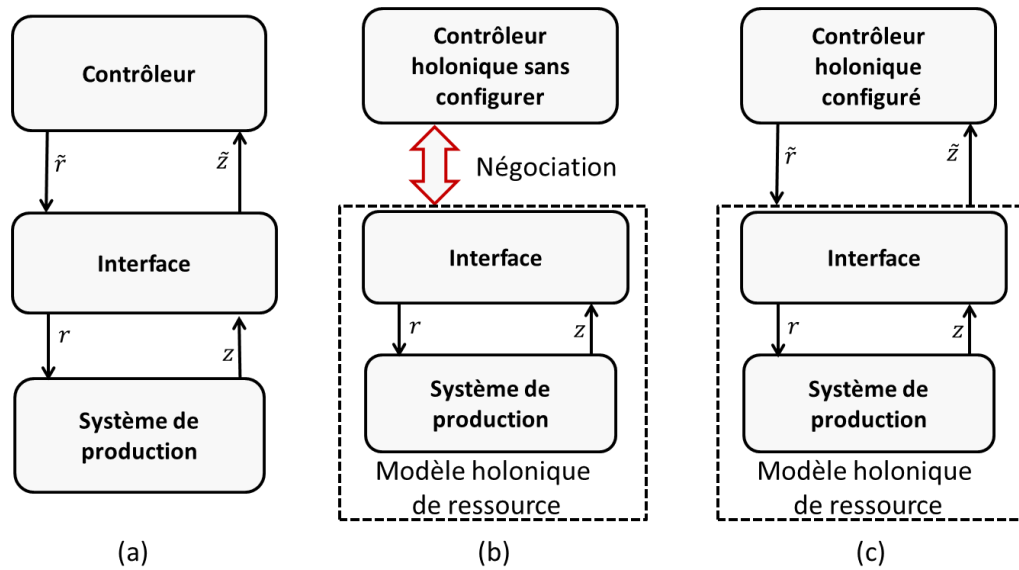


Figure 19. (a) Système de contrôle hybride, (b) Représentation holonique pendant le process de planification, (c) Représentation holonique pendant le process de supervision.

Ce chapitre passe en revue une introduction des systèmes de contrôle hybrides, depuis des définitions de base de systèmes hybrides jusqu'au modèle hybride de (Antsaklis et al., 1993). Le problème de flexibilité sur les systèmes de contrôle hybrides traditionnels a particulièrement été mis en avant.

Dans ce chapitre a aussi été présentée une introduction des systèmes holoniques de façon générale, finissant par l'explication de l'architecture de référence holonique PROSA, et comment la caractéristique de flexibilité de ce type de systèmes est intéressante pour les HCS.

Le chapitre suivant introduit notre contribution principale, nommée H²CM, adaptation de l'architecture PROSA pour la modélisation des HCS.

Chapitre 2

2 Application du paradigme holonique aux systèmes hybrides

L'utilisation du paradigme holonique pour la modélisation des systèmes de contrôle hybrides n'est pas totalement novatrice. En effet, plusieurs travaux existent dans la littérature et ont déjà traité de cette possibilité. De plus, plusieurs travaux connexes et très proches existent également, concernant majoritairement l'utilisation des systèmes multi-agents. La première section de ce chapitre présente une revue détaillée de ces différents travaux. La majeure remarque faite sur ces travaux est qu'ils sont généralement basés sur une étude de cas industriel. Cette spécificité permet d'appuyer l'évaluation de performance de la solution proposée, mais nuit à la généralisation des travaux autour de définitions plus génériques des holons de l'architecture proposée.

La seconde section de ce chapitre introduit alors le cœur de la proposition de ce document, en l'occurrence une définition d'architecture générique permettant la modélisation de tous types de systèmes de contrôle hybride et s'extrayant du contexte classique de ce type de systèmes.

2.1 Travaux antérieurs

Cette section propose un focus sur une sélection d'articles de la littérature utilisant ou se rapprochant du paradigme holonique dans un contexte de modélisation de systèmes de contrôle hybrides. La littérature complète est relativement limitée, mais ces articles montrent suffisamment de détails sur la modélisation proposée des agents/holons pour pouvoir servir de base pour un développement d'architecture générique dans la seconde section de ce chapitre.

2.1.1 Approches multi-agent

Tout d'abord, la littérature présente deux travaux majeurs utilisant les systèmes multi-agents pour le contrôle de systèmes continus. Ces travaux mettent en évidence les caractéristiques d'autonomie et de négociation permettant de doter le contrôle de systèmes hybrides de la flexibilité recherchée ici.

2.1.1.1 Systèmes de contrôle basés sur les systèmes Multi-agents : une approche hybride pour le contrôle de process distribués

(Iglesias et al., 1996) ont proposé une architecture multi-agent permettant d’englober une multitude d’agents hétérogènes, et ainsi créer des architectures de contrôle hybrides. Ces résultats découlent du projet ESPRIT-9119 financé par la Commission Européenne et dénommé MIX (Modular Integration of Connectionist and Symbolic Processing in Knowledge Based Systems - (Hilario et al., 1994)), incluant des laboratoires d’intelligence artificielle français, suisses, allemands et espagnols (INRIA-Lorraine, France; CUI-University of Geneva, Switzerland; IMAG-Grenoble, France; Kratzer Automation, Munich, Germany; Technical University of Munich, Germany; et Technical University of Madrid, Spain). L’architecture de même nom proposée a été conçue en accord avec les principes suivants :

- Utilisation des mécanismes d’encapsulation, d’isolation et de contrôle local : chaque agent est une entité autonome et semi-indépendante ;
- Aucune hypothèse n’est faite quant à la connaissance individuelle de chaque agent ou leur méthodes de résolution de problèmes ;
- Une organisation flexible et reconfigurable dynamiquement est autorisée.

Chaque agent (Figure 20) est composé d’un bloc de contrôle, d’une base de données (incluant un modèle de l’agent, un modèle de l’environnement, l’état de l’agent et quelques données privées et/ou globales) et d’un bloc de communication (incluant le modèle du réseau de communications et une boîte aux lettres).

Tout agent peut avoir des objectifs propres (en pratique des processus qui démarrent quand l’agent est créé), des services offerts (l’agent offre un ensemble de services au reste des agents, services qui peuvent être exécutés de manière concurrente ou non) et des services requis (une liste contenant les noms des services dont cet agent peut avoir besoin).

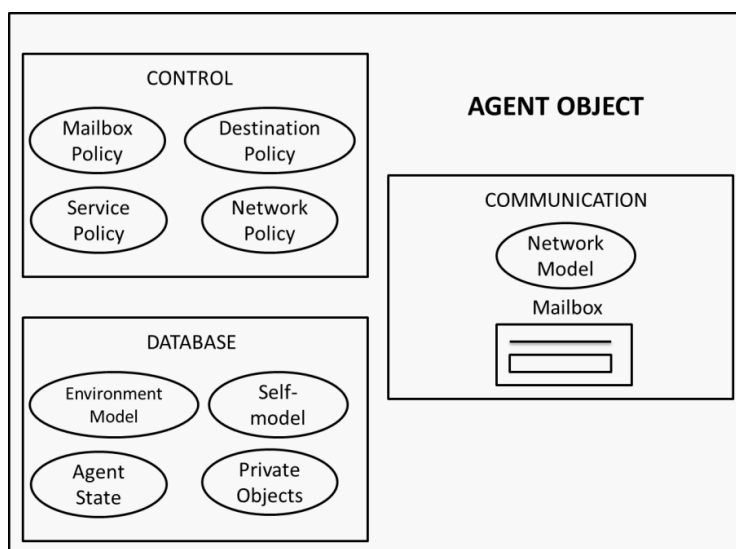


Figure 20. Modèle d’un agent multiservice

Une des caractéristiques principales de ces agents est que leurs services (s'ils sont concurrents) sont exécutés en tant que process séparés, donc l'algorithme de contrôle de l'agent peut continuer son office. De cette manière, le même service (concurrent) peut être exécuté plusieurs fois, chacune appelée depuis un agent différent.

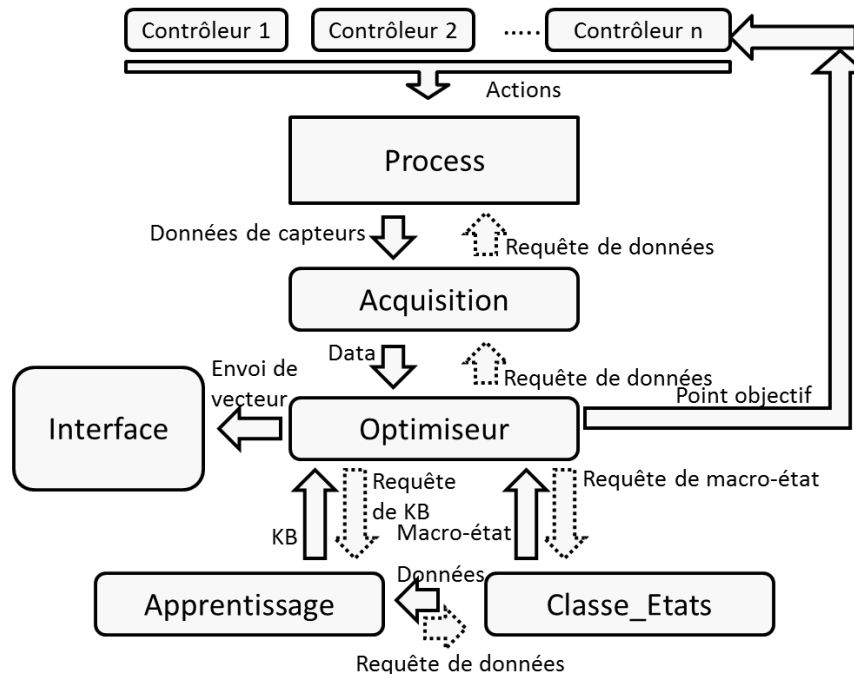


Figure 21. Description des agents

Pour la conception d'applications utilisant la plateforme MIX, ce système de commande distribué doit être considéré comme un ensemble d'agents ayant leurs objectifs et services respectifs, communiquant entre eux à travers l'échange de messages. La Figure 21 présente une description graphique de ce système, où chaque action principale ou groupe d'actions peut être vu comme un agent ayant plusieurs objectifs/services.

L'agent Acquisition obtient des données des capteurs situés à l'intérieur du process de manière périodique, et fournit sur demande des « vecteurs de contexte » contenant l'état relevé du système à l'Optimizer. L'agent Optimizer demande à l'agent Class_state le macro-état approprié, et utilise la ou les bases de connaissance appropriée(s) pour obtenir un « vecteur opérationnel ». Le contenu des variables de ce vecteur est envoyé aux contrôleurs spécifiques à certains points d'actions du système, ou affiché aux éventuels opérateurs en cas de nécessité d'ajustements manuels. L'agent Optimizer requiert au système d'apprentissage Learning la constitution d'une nouvelle base de connaissances si la valeur du coût (dérivé indirectement du flux de chaleur) est détectée en augmentation.

2.1.1.2 Systèmes de contrôle hybride et autonome par de multiples agents

Ces travaux sont détaillés dans (Kohn and Nerode, 1992) et (Nerode and Kohn, 1993b). L'architecture de contrôle hybride et autonome par de multiples agents (Multiple Agent Hybrid Control Architecture) est un framework général pour l'implémentation temps-réel de contrôleurs distribués, réactifs et intelligents. Le mécanisme central de l'architecture

est un mécanisme de preuve de théorème en ligne utilisant une théorie des formes rationnelles. L'architecture est composée d'une collection d'agents et d'un réseau interconnectant tous ces agents. L'architecture est spécifiquement adaptée pour les systèmes impliquant de multiples centres de décisions sans protocole strict de prise de décision spécifié a priori. L'exemple spécifique choisi dans ces travaux est celui des systèmes de contrôle d'ateliers manufacturiers.

Dans la plupart des modèles proposés pour les systèmes hybrides, les composants continus sont représentés par des équations différentielles et les composants discrets par des automates à événements finis (Nerode and Kohn, 1992). Dans ces modèles, chaque agent modélise l'évolution du système avec un ensemble d'équations différentielles couplées, un modèle logique avec un ensemble de bases de connaissances (Mettala and Graham, 1992) et un ensemble de mécanismes et principes d'inférence pour déduire les actions de contrôle locales à chaque date de décision en tant que fonction objectif, données de capteurs et instance actuelle de la base de connaissances. Les fonctions centrales de chaque agent sont :

- Construire en ligne un automate, appelé automate d'inférence ;
- Simuler son comportement pour pouvoir générer les actions de contrôle qui satisfont les contraintes fonctionnelles et lui permettent de tendre vers son objectif ;
- Adapter le comportement du système en cas de défaillance.

Chaque agent de l'architecture de contrôle hybride est pourvu d'un prouveur automatique de théorème temps-réel basé une théorie variationnelle relaxée, développée par (Mesarovic et al., 1970). Un agent de contrôle est composé de cinq éléments fonctionnels, comme illustré Figure 22 :

- Une base de connaissance ;
- Un planificateur de théorème ;
- Inférenceur, qui est un démonstrateur de théorème équationnel en ligne ;
- Un décodeur de données ;
- Un adaptateur.

L'architecture construite suit un ensemble de caractéristiques qui sont :

- Temps-réel : les contraintes de performance temps-réel sont explicites et font partie de la base de connaissances ;
- Réactive : la fonction de preuve de théorème de chaque agent de l'architecture opère en boucle fermée du premier ordre ;
- Adaptative : la base de connaissances de chaque agent est ouverte et modifiable par les données issues des capteurs. Une défaillance liée au théorème implique automatiquement un réglage ou une correction de la base ;
- Distribuée avec coordination : la fonction de preuve de théorèmes est distribuée entre tous les agents. Le mécanisme de coordination est implicite sans arbitrage ;

- À hiérarchie dynamique : l'architecture peut opérer simultanément à plusieurs niveau d'abstraction ;
- Facteur de mérite : le comportement en boucle fermée du système distribué est déterminé en prouvant qu'il existe au moins une trajectoire de contrôle qui minimise une fonction objectif donnée.

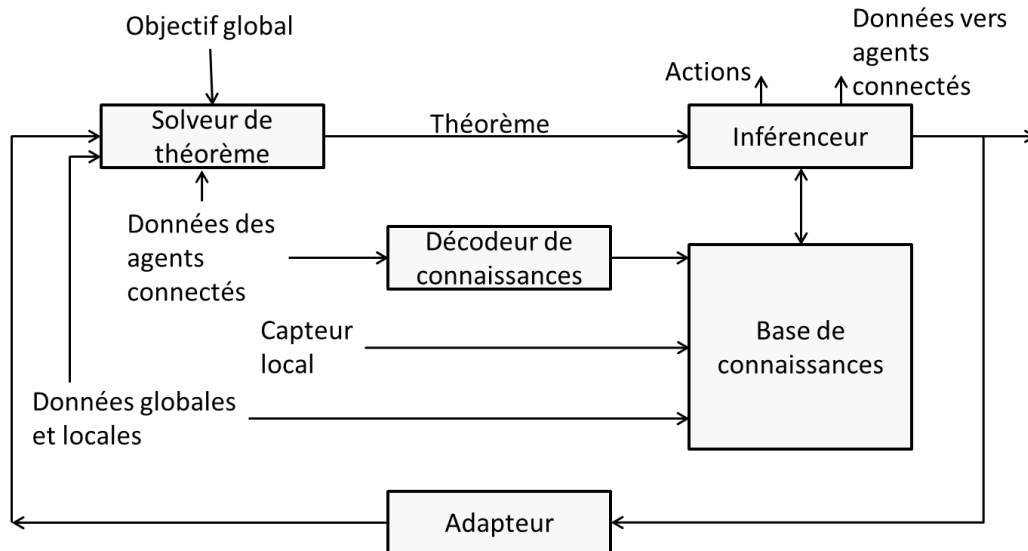


Figure 22. Structure multi-agent

2.1.1.3 Conclusion sur les travaux orientés multi-agents

Les agents ont des caractéristiques d'autonomie et de négociation qui sont intéressantes dans l'application aux systèmes hybrides. Les agents sont utilisés dans ces travaux comme un outil de contrôle du process continu. Cette étude de l'application des systèmes multi-agents aux systèmes hybrides montre que l'utilisation de la base de connaissances et des interfaces de communication est une base solide pour la création des mécanismes d'autonomie et de coopération. Ces travaux montrent aussi comment les agents peuvent interagir avec les systèmes continus, à travers des mécanismes d'acquisition de données.

Lorsque l'on étudie plus en détail les architectures de contrôle de ces travaux, l'on peut s'apercevoir que les agents construits ont des fonctionnalités relativement semblables mais que le périmètre d'action de chacun est relativement différent et que la terminologie utilisée n'est pas du tout uniforme. L'utilisation du paradigme holonique a pour objectif d'aider à la conception d'architectures de contrôle en donnant un cadre de conception plus contraignant. La section suivante présente les travaux de la littérature utilisant le paradigme holonique pour la conception des architectures de contrôle de systèmes hybrides.

2.1.2 Approches holoniques

Trois applications des systèmes holoniques sur des systèmes continus sont présentées dans cette section. Notre objectif est de présenter comment ces travaux définissent les fonctions de chaque élément de l'architecture afin de généraliser l'approche dans la fin de ce chapitre.

2.1.2.1 Systèmes de production holoniques à process continu: concepts et exigences de contrôle

Une implémentation de système de production holonique sur un système continu est par exemple proposé par (McFarlane, 1995). La Table 1 montre la problématique de modélisation du système en fonction du niveau d'abstraction du système de contrôle. Bien que les algorithmes et les contraintes soient clairement différents entre les niveaux, chaque cas induit une régulation de la valeur d'une variable de commande via une commande bouclée de manière à aligner la valeur d'une variable de sortie sur une valeur de consigne prédéfinie.

Table 1. Problèmes de contrôle statiques et dynamiques à différents niveaux de contrôle

Niveau d'abstraction	Type	Phase statique	Phase dynamique
Contrôle local	Continu	Mise au point basée sur les modèles	Contrôle basé sur les modèles
Contrôle du process	Discret / Continu	Ordonnancement	Réordonnement dynamique
Management du process	Discret	Planification	Planification dynamique

Le système décrit Figure 23 a la caractéristique de convertir en interne les objectifs et exigences en un ensemble acceptable de paramètres ou trajectoires de comportement via une fonction de décision D. L'action de contrôle calculée est ensuite appliquée sur le processus P. La terminologie utilisée dans la Figure 23 est la suivante :

- P - process (+ modèle + contrôleur)
- D - Problème de prise de décision
- ϕ - Principe de prise de décision
- M – ensemble de décisions alternatives
- U – ensemble d'évènements alternatifs
- Y – ensemble de sorties possibles
- $m \in M, u \in U, y \in Y$
- $P: M \times U \rightarrow Y$

Le développement de problèmes de décision appropriés (et flexibles) est la clé pour inclure les activités de contrôle dans un système holonique dont les objectifs peuvent varier en fonction de l'environnement dans lequel il opère. Une prise de décision distribuée basée sur des algorithmes d'optimisation est bien adaptée aux systèmes holoniques, et les méthodes de recherche d'objectifs peuvent aider à l'intégration d'algorithmes de contrôle basés sur des mécanismes d'optimisation complexes dans des process manufacturiers, ce qui est relativement rare jusqu'à présent.

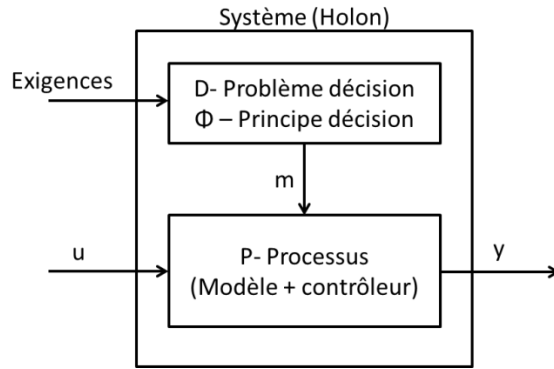


Figure 23. Framework de contrôle

La Figure 24 montre qu'un algorithme de type H^∞ est naturellement un bon candidat pour s'insérer dans ce framework. La prise de décision, f , est la minimisation de la norme H^∞ d'une fonction de coût, généralement notée :

$$\phi = \inf_{K \text{ stabilising}} J_\infty = \inf_{K \text{ stabilising}} \| [F_L (M(G, W), K)] \|_\infty$$

Où :

- G est un modèle du process ou une fonction à contrôler ;
- W est la fonction de pondération associée à la fonction de coût ;
- M est appelée l'usine standard H^∞ , combinant les pondérations, le modèle du process et ses interconnexions ;
- K est le contrôleur à concevoir ;
- et $F_L(\bullet)$ est une Transformation Fractionnelle Linéaire combinant ces éléments sous la forme de la fonction coût.

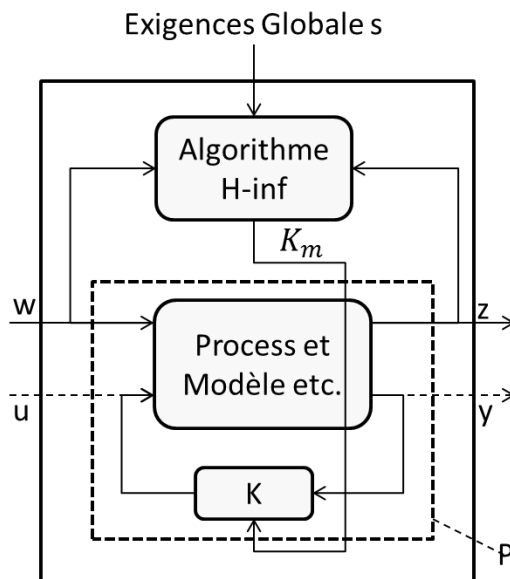


Figure 24. Algorithme de type H^∞ dans le Framework à recherche d'objectifs.

Le problème de prise de décision D revient alors à sélectionner les pondérations appropriées pour l'optimisation, afin de refléter au mieux les exigences globales et déterminer un ensemble approprié de contrôleurs aptes à satisfaire ces exigences. Ainsi, si l'on se réfère de nouveau à la Figure 23, le process P encapsule le process réel, un modèle de ce process et le contrôleur sélectionné par D.

Par exemple, l'objectif global du contrôle d'un holon au niveau local ou process peut être d'atteindre une consigne dans un intervalle d'1% en régime permanent (avec des éventuelles fréquences d'oscillation inférieures à 0.01 rad.s^{-1}), avec la contrainte additionnelle que le modèle du process (noté G) inclus dans l'holon ne peut garantir une précision inférieure à 10% du comportement réel du process à haute fréquence, i.e. au moment où plusieurs modes dynamiques sont négligés. La prise de décision doit alors fournir un contrôleur K_m issu de la classe des contrôleurs possibles K qui permettra d'atteindre cet objectif. Mathématiquement, ce problème spécifique est référencé comme un problème de sensibilité mixte. Il est à noter que le mécanisme de prise de décision concernant la pondération de la fonction coût fait intervenir des systèmes experts et est partie intégrante de la fonction de contrôle à recherche d'objectifs.

Les holons ont également besoin de capacités de diagnostic de manière à pouvoir mettre en œuvre un comportement autonome. Les deux aspects majeurs à définir sont les capacités d'un holon à détecter ou diagnostiquer les problèmes en interne, ou en coopération avec des holons voisins si nécessaire. La détection implique les fonctions d'identification mais pas forcément la fonction de traçage d'un problème au sein du système complet. La méthode proposée pour un environnement basé sur le paradigme holonique utilise des bilans de santé soumis par chaque holon pour indiquer un fonctionnement normal ou d'éventuels problèmes. Ce bilan de santé est la sortie d'un algorithme local de détection de défaillances à l'intérieur du holon, qui peut être implémenté au travers d'une large gamme d'instruments ou d'approches de redondance analytique (basée sur les modèles), chacune d'entre elles étant fondamentalement une comparaison entre le comportement réel et le comportement nominal prédit par le modèle du système. Tous les holons calculent ainsi leur bilan de santé et soumettent leurs statuts à un holon de diagnostic local. Le bilan de santé est généralement exprimé comme une valeur issue d'un intervalle entre 0 et 1 (relative à la différence entre réel et nominal), où 1 indique un comportement normal et 0 indique une défaillance complète du système. Au moment de la détection d'une défaillance, le holon de diagnostic initie une procédure d'interrogation des bilans de santé de chaque holon afin de constituer une signature de la défaillance et si nécessaire étendre cette procédure aux holons des niveaux hiérarchiques inférieurs.

La caractéristique qui nous intéresse dans ce travail est le cadre de contrôle proposé, dans lequel un algorithme est utilisé dans la prise de décision du Holon sur le process continu avec contrôleur continu. C'est une architecture de contrôle très simple, qui n'a aucun Holon de type ordre ou de type produit. Elle offre toutefois un mécanisme de planification par appel d'offres et un mécanisme de supervision en utilisant le concept d'état de santé.

2.1.2.2 Architecture d'intégration pour l'automatisation d'un complexe de production continu.

Une autre architecture holonique développée pour des systèmes continus est celle proposée par (Chacón et al., 2002). Une structure basée sur un système de prise de décision appelé Unité de Production (PU) est proposée. Cette unité est le fondement pour exploiter la structure fractale caractéristique des systèmes holoniques. L'agrégation des PU est alors appelée Système de Production Complexe (SPC). Les PU contrôlent en interne leur propre comportement en fonction de leurs objectifs propres. Le système agrégé cherche à atteindre ses objectifs en assignant des objectifs dérivés à chaque PU. Une PU est composée d'un ensemble d'unités élémentaires (équipements), qui sont organisées de telle manière qu'elles permettent de réaliser des processus de transformation (chimique, physique, de transport, etc.), afin d'obtenir les produits finaux avec de bonnes performances en terme de Coût/Qualité/Délai. Si une PU contient plusieurs sous-systèmes, le comportement optimum est établi suivant la même voie hiérarchique. De ce fait, l'organisation de l'architecture des SPC peut être structurée en couches, similaire au modèle classique CIM (Scheer, 2012).

Dans ce contexte, les trois éléments suivants sont utilisés pour décrire les SPC: holons Ressources, holons Mission-Produits et holons Ingénierie.

- **Ressources.** Les ressources peuvent être de plusieurs types :
 - Infrastructure de Production : Cette ressource est associée aux éléments où les fonctions de transformation du produit sont réalisées. L'infrastructure a une capacité nominale selon sa conception, et une capacité réelle en fonction de son historique de fonctionnement.
 - Produits bruts : Cette ressource a une certaine qualité, qui influence la qualité du traitement.
 - Personnel : Associé à la production, la capacité et la disponibilité des ressources humaines est indispensable pour définir l'exécution des tâches de contrôle, supervision, mesure, planning, etc.
 - Services : Autres ressources comme l'énergie, la réfrigération, l'eau non souillée, etc. qui peuvent être nécessaires pour le processus.
- **Mission-Produits :** À chaque PU est assigné un ensemble de produits obtenus en utilisant les ressources et leurs méthodes de production. La mission d'une PU est d'obtenir un lot de produits dans un intervalle de temps prédéfini. Chaque PU a également un ensemble prédéfini d'objectifs qu'il convient d'évaluer vis-à-vis de l'évolution de la PU. Dans le cas de processus de production continu, le produit est associé à des objectifs de production, mesurés en termes de quantité par unité de temps.
- **Ingénierie :** Une production est développée en utilisant des ressources et suivant des méthodes de production spécifiques. Il existe une méthode optimale pour accomplir une production dépendant des ressources disponibles au cours du temps et respectant les spécifications prédéfinies du produit. Cette méthode peut évoluer au cours du temps lorsque des changements interviennent, tant au niveau du

produit semi-fini, des équipements ou des spécifications du produit en lui-même. Le holon Ingénierie contient toutes les spécifications du produit.

La dynamique d'une PU peut être représentée par des formalismes à transitions tels que les réseaux de Petri ou les automates à états finis. Le comportement de tels systèmes peut être décrit en termes de systèmes à événements discrets (SED). De cette façon, il est possible de suivre les changements dans les ressources ou missions et, du point de vue de l'ingénierie, de sélectionner de nouvelles recommandations pour les objectifs de production. Les relations entre les trois éléments d'une PU sont présentés Figure 25.

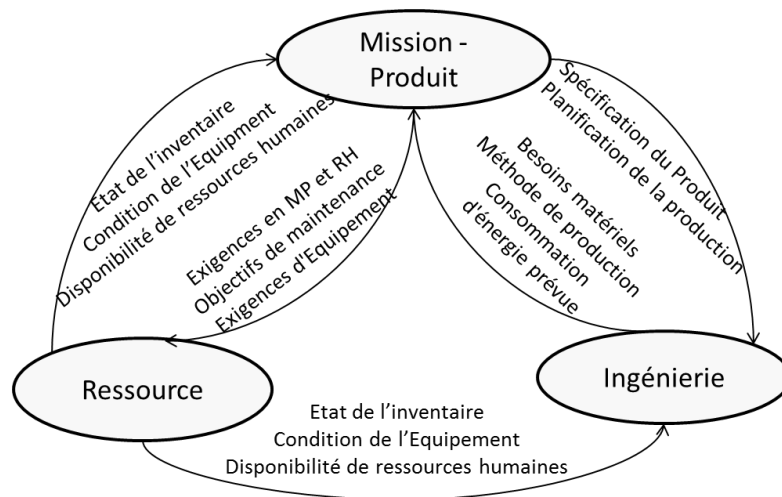


Figure 25. Relations entre les trois éléments d'une PU

De manière à intégrer les systèmes d'automatisation des SPCs, il est nécessaire de modéliser les PU. Une PU est décrite comme un système autonome comme défini dans (Antsaklis and Passino, 1993). Selon cette vision, une PU est apte à prendre des décisions en interne concernant ses missions, telles que : changer la procédure de réalisation des produits, détecter si l'un de ses objectifs a été modifié et informer si le nouvel objectif est atteignable... Elle doit aussi informer lorsqu'un objectif est accompli ou s'il ne peut être atteint du fait de défaillances de la PU. Matières premières, énergie, services et informations sont les entrées d'une PU, tandis que les produits finis, de l'information sur le suivi de production et les déchets sont ses sorties, comme illustré sur la Figure 26.

Deux éléments principaux coexistent au sein d'une PU. En premier, nous pouvons citer les éléments physiques (e.g. mécaniques, électriques, etc.) et/ou les process chimiques apportant une transformation du produit. En se basant sur la connaissance contenue dans la PU, nous pouvons également citer les systèmes d'information et de prise de décision, qui permettent le contrôle et la régulation des process. Ils échangent aussi de l'information avec d'autres PU de manière à garantir le contrôle du système complet.

La description d'une PU peut être structurée en objets internes et externes, qui peuvent créer ou modifier les missions confiées à la PU. Les objets internes d'une PU peuvent être listés comme :

- **Type de produit** : Liste de produits finis pouvant être obtenus par un process au sein d'une PU en utilisant une méthode de production donnée ;
- **Méthode de production** : Associée à chaque type de produit, une ou plusieurs méthodes de production dépendent du produit brut et de l'état de l'équipement ;
- **Mission** : Une mission est reliée à la production à réaliser au sein d'un créneau de temps donné. Pour réaliser cette mission, l'opération d'une PU doit être divisée en un ensemble de tâches à réaliser par l'infrastructure technique (équipement ou d'autres PU plus élémentaires) ;
- **Capacité de production** : Plus haute cadence de sortie pouvant être atteinte pour un type de produit dans des conditions idéales.
- **Etat de l'équipement** : Une PU est composée d'une ou plusieurs infrastructures qui réalisent les transformations physiques et/ou chimiques des produits. Ces infrastructures, au cours de leur fonctionnement, évoluent en performance en fonction de l'historique de leur utilisation. Cet objet indique donc l'état de chaque infrastructure, d'où l'on va déduire une capacité réelle de performance.
- **Etat du process** : Evolution du process pour réaliser un type de produit donné, résultant de la composition des capacités des outils de production, de l'état des équipements (capacité réelle) et de l'avancement des missions en utilisant une méthode de production.

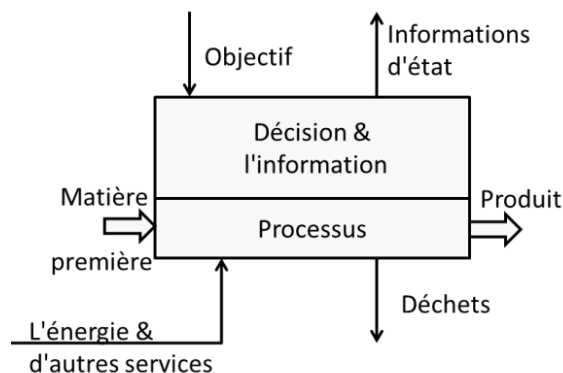


Figure 26. Unité de Production

Pour résumer, l'architecture proposée ici est basée sur une structure de prise de décision appelée unité de production. Cette structure ressemble à la structure proposée précédemment dans (McFarlane, 1995), mais les rôles et les types de holons sont définis différemment. Une caractéristique qui nous intéresse particulièrement au sujet de ce système est son mécanisme de négociation, exprimé Figure 25.

2.1.2.3 Une approche de coordination distribuée pour un process de contrôle reconfigurable

Les travaux développés par (Chokshi and McFarlane, 2007) proposent une approche de coordination de process distribués dans le contexte de contrôle reconfigurable dénommée DRPC (Distributed Reconfigurable Process Control). Ainsi, l'architecture de contrôle est développée en utilisant quatre éléments de base que sont le Produit, l'Unité, l'En-tête et le

Service. Bien que ces travaux n'adressent pas spécifiquement le paradigme holonique, nous le citons car le résultat est très proche de ces considérations, sans les nommer. Le but de leur architecture est d'obtenir un système de contrôle basé sur des modèles d'interaction reconfigurables et flexibles. Leur système est conçu pour pouvoir être implémenté à la fois sur des process continus ou semi-continus. Les fonctionnalités de chacun des éléments peuvent être explicitées de la manière suivante :

- **Unité** : Tâche de transformation physico-chimique telle qu'une réaction, distillation etc. au sein du process. Peut être associée à au moins un mais potentiellement plus d'une décision du système de contrôle que l'unité peut réguler en propre ;
- **En-tête** : Représente la logistique des matériaux ou services au sein d'un segment spécifique du process complet. Physiquement, il peut correspondre à un certain nombre de réseaux de tuyauteries, d'équipements de transfert (pompes, compresseurs etc.), d'éléments de contrôle qualité, d'unités d'échange de chaleur ou de stockage. Ces composants se différencient des unités car ils n'apportent pas de transformations physico-chimiques, bien qu'ils puissent avoir une influence sur certaines propriétés physiques telles que le débit, la chaleur, la pression, etc.
- **Service** : Un fournisseur de service représente un composant en charge d'allouer un service à des éléments de process clients utilisant ce service dans leur tâche. Les éléments clients peuvent être soit des unités soit des en-têtes. Plusieurs fournisseurs peuvent exister dans le système, chacun fournissant un ou plusieurs services à la fois.
- **Produit** : Représente les contraintes de production d'un ordre de fabrication donné sous la forme d'une gamme de production et d'autres spécifications telles que la qualité, la quantité et la cadence du produit demandé. De multiples produits peuvent co-exister dans un même process, chacun représentant un ordre spécifique, mais seulement quelques-uns peuvent être produits à la fois.

L'approche DRPC a pour objectif *de minimiser le couplage* entre les éléments du système de contrôle afin de faciliter la reconfiguration. Cela permet d'un côté de construire une architecture distribuée comprenant des composants modulaires ayant leur propres capacités de prise de décision, et d'un autre côté un mécanisme de coordination qui retarde au maximum le couplage entre ces composants et leur utilisation dans l'atelier, où l'état réel de l'atelier doit être pris en compte. Un exemple classique de ce dernier point est le découplage des informations de gamme de réalisation des produits de l'utilisation des équipements de l'usine. Les informations de gammes sont intégrées aux capacités de l'atelier au dernier moment via des interactions dynamiques et distribuées entre les éléments de contrôle.

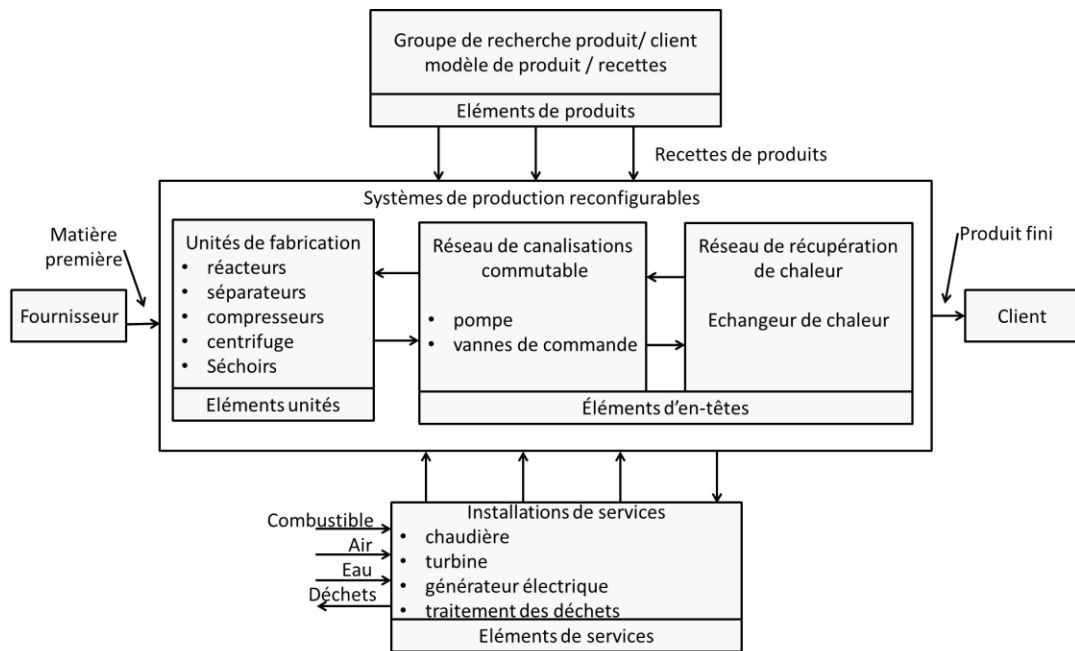


Figure 27. Fonctionnalités des éléments basiques de contrôle du process DRPC

Un modèle d'interaction a été proposé pour la phase de reconfiguration du système (Figure 28) et est structuré en cinq étapes : Identifier, Définir, Reconfigurer, Opérer et Terminer.

- **Identifier** : La reconfiguration commence par l'identification d'une possibilité de production (généralement lorsqu'un ordre de fabrication arrive) ou par la décision de modification d'un ou de plusieurs process en cours actuellement sur le système (détection d'une perturbation quelconque) ;
- **Définir** : L'étape de définition est décomposée en deux sous-phases: (i) *recipe mapping*, pour associer les informations contenues dans la gamme du produit avec les capacités de production disponibles dans l'atelier, et (ii) *synthèse*, pour déduire un process à partir des choix potentiels créés par le mapping.
- **Reconfigurer** : Cette étape peut être conceptuellement décomposée en trois sous-phases: découpler, réorganiser et réaccoupler. Chacune de ces sous-phases peut inclure un changement du système physique ou de son contrôle.
- **Opérer** : Les unités commencent le traitement du produit de manière coordonnée afin de transformer les stocks d'alimentation en produit fini. La coordination de ces activités aboutissant au démarrage et aux différentes opérations de process en découlant est réalisée par les unités elles-mêmes.
- **Terminer** : La configuration du process est finalement dissolue une fois que l'ordre concerné est considéré terminé ou si une défaillance majeure apparaît.

Le concept de tâche est central dans ces travaux. C'est un élément qui définit les fonctions des ressources (appelées services dans ce document). La planification au sein de cette architecture se fait à travers la négociation entre ressources (unités), sur la base des informations du produit sans recourir à un élément spécifique de surveillance des ordres de production.

2.1.3 Discussion

Nous avons identifié plusieurs travaux antérieurs majeurs ayant utilisé une architecture holonique sur des systèmes continus. Une étude de ces travaux a permis de relever plusieurs caractéristiques importantes qu'une architecture générique se doit d'avoir. La Table 2 présente un comparatif de ces travaux sur la base des fonctionnalités attendues de l'architecture générique.

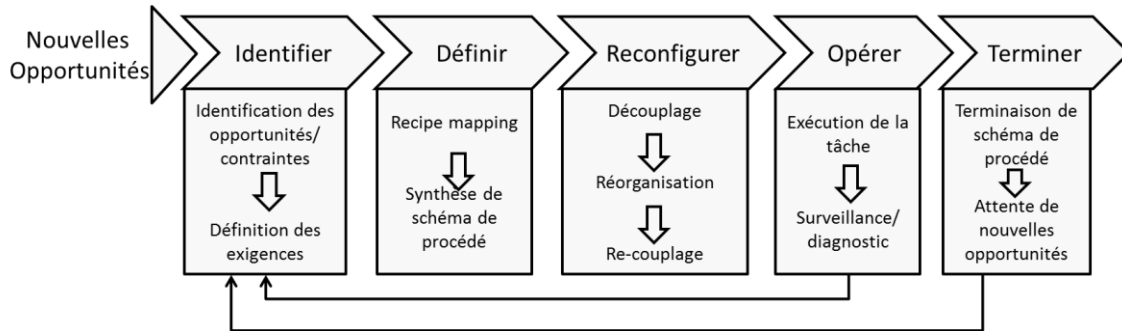


Figure 28. Distributed reconfiguration process

De manière générale, ce que l'on peut observer est que ces approches ont été bâties à partir d'exemples industriels. Le résultat obtenu permet de fait de démontrer la faisabilité de l'approche, mais a le défaut de diriger la conception de l'architecture à partir des données du terrain. Ainsi, les éléments des architectures de contrôle proposés sont très liés aux fonctions qu'ils doivent avoir, ce qui implique un manque de généricité dans l'application de ces architectures sur des problèmes différents. Cela déconnecte de plus les résultats de ces travaux de l'ensemble des résultats déjà disponibles des nombreuses études issues de la communauté des systèmes à évènements discrets.

Pour chaque travail, nous avons néanmoins identifié un point fort qui sera la base des développements présentés dans la section suivante.

(McFarlane, 1995) propose un framework de contrôle, dont la structure inclut le process, l'algorithme de prise de décision et l'application de cette décision. De plus, toute l'information manipulée par l'algorithme est obtenue via un système d'acquisition de données directement sur le système.

(Chacón et al., 2002) propose l'architecture proposant le plus grand découplage entre les holons du système. Comme indiqué précédemment, c'est ce découplage qui permettra d'optimiser le process de conception mais surtout de reconfiguration du contrôle du système.

Enfin, (Chokshi and McFarlane, 2007) base tout son travail sur le principe de tâches de production. Cette notion est analogue à celle de services que l'on peut retrouver en informatique. Dans nos travaux, nous proposons de nous baser sur le concept de services afin de définir les gammes de produits et capacités des ressources.

Table 2. Tableau comparatif des travaux antérieurs.

Éléments de l'architecture	(McFarlane, 1995)	(Chacón et al., 2002)	(Chokshi and McFarlane, 2007)
Modèle de ressource	Tout élément du système est une ressource	Unité de production, représentant les machines.	Divisés en éléments Unités, En-têtes et fournisseurs de service
Modèle de produit	N/A	Holon Ingénierie, contenant les informations de gamme des produits.	Élément Produit, contenant les informations de gamme des produits.
Modèle d'un ordre	N/A	Holon Mission-Produit : Chaque unité inclut un ensemble de produits obtenus en utilisant les ressources et méthodes de production.	N/A
Récurtivité / Fractalité	Non	Les PU sont structurées récursivement au sein de couches qui sont ensuite superposées	Non
Planification	Etablissement de contrats entre ressources	Négociation entre holons au sein d'une même couche ou entre plusieurs couches	De façon centralisée, chaque ressource demande sa configuration à l'holon Produit. De façon distribuée, une négociation est établie entre les Ressources et les Produits en partant de la fin du process
Supervision	Création du concept de bilan de santé	Utilisation du système d'aide à la décision des PU, en comparant les données de capteurs à une fonction objectif	N/A

2.2 Holonic hybrid control model

L'application du paradigme holonique aux systèmes hybrides démarre systématiquement par la même interrogation: Quel élément du système peut être considéré comme étant le produit? En effet, l'approche orientée événements discrets est focalisée sur les produits et sur les traitements que les différentes machines et humains opèrent sur ceux-ci (Pannequin et al., 2009; Sallez, 2014; Yang et al., 2009). La nature des produits considérés par les systèmes de contrôle hybrides ne permet pas l'application de ces mêmes concepts. De plus, la problématique d'inter-connectivité est différente: soit la production se fait par lots (batch production), ce qui peut être vu comme une production discrète en containers, soit par l'intermédiaire d'un réseau de tuyauterie, où c'est un ensemble de capteurs qui renseignent sur l'évolution du produit. En se concentrant sur le seul comportement hybride, ce sont donc les machines qui sont vouées à prendre en charge la prise de décision, et non les produits eux-mêmes.

Cette section introduit l'application d'une architecture holonique générique aux HCS, comme illustrée Figure 29. Les éléments constitutifs des HCS, contrôleur, interface et système de production sont reformulés dans le modèle de contrôle hybride holonique (Holon Hybrid Control Model – H²CM). Le contrôleur est représenté par les holons Ordre et Produit, l'interface est représentée par la partie logique des holons Ressource et le système de production est représenté par la partie physique des holons Ressource. Pour cela, les holons Produit, Ressource et Ordre sont définis et un modèle conceptuel d'orchestration est proposé. Puis, le mécanisme principal de négociation est détaillé de manière à illustrer la chorégraphie des holons, pour finir par la description succincte d'un système de supervision basique.

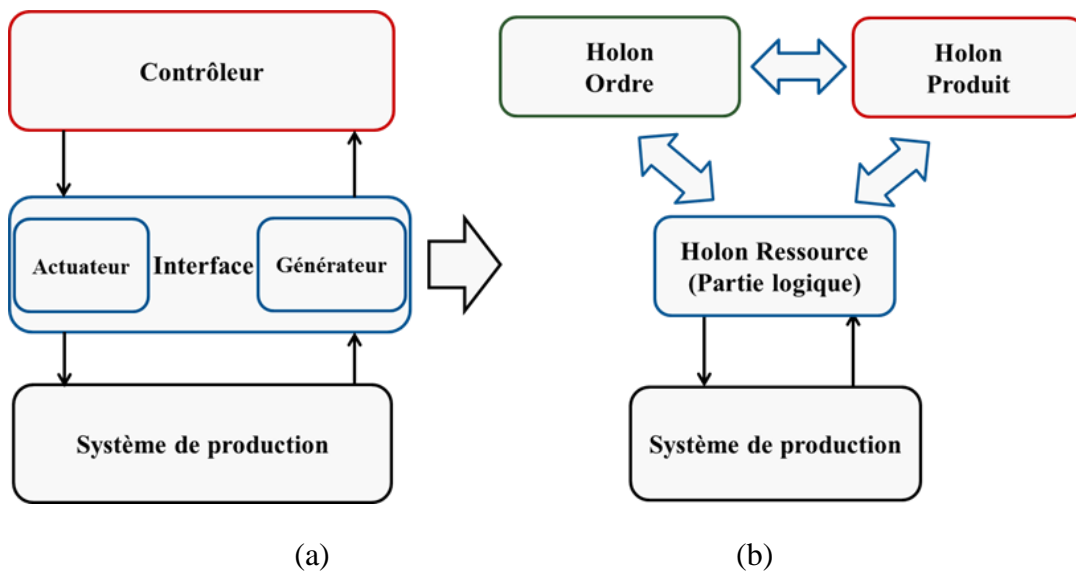


Figure 29. Du HCS (a) au H²CM (b).

2.2.1 Structure fractale de H²CM

Comme indiqué précédemment, H²CM est basé sur PROSA, et plus précisément sur ses trois holons basiques. Le holon Staff n'est pas considéré ici, car son rôle est extrêmement dépendant du contexte et difficile à exprimer de façon générique. De plus, son objectif dans H²CM reste globalement le même que dans une configuration discrète, donc l'hypothèse faite dans ces travaux est de considérer que son modèle et son comportement restent les mêmes que dans un système holonique classique. La Figure 30 introduit le diagramme de blocs élémentaires de H²CM. Deux différences majeures avec la modélisation classique des holons dans PROSA (Figure 14) apparaissent :

1. Chaque ressource est dotée d'un Ordre et d'un Produit tout au long de son existence. Le holon Ordre est en charge de la surveillance de la ressource tandis que le holon Produit est en charge de la gamme à appliquer au produit actuellement en fabrication. Le contenu et les objectifs de ces holons Ordre et Produit évoluent constamment pour s'adapter aux changements durant la production, mais cette structure reste en permanence la même;
2. Un lien de récursivité est présent au niveau du holon Ressource. En effet, il peut exister des ressources composées, qui peuvent être décomposées en une ou plusieurs holarchies, comprenant une ou plusieurs ressources et leurs holons Ordre et Produit associés. De manière similaire à la structure précédente, les relations d'agrégation créées ici peuvent changer au cours de l'exécution du système, i.e. des holarchies peuvent être créées, modifiées ou détruites en ligne.

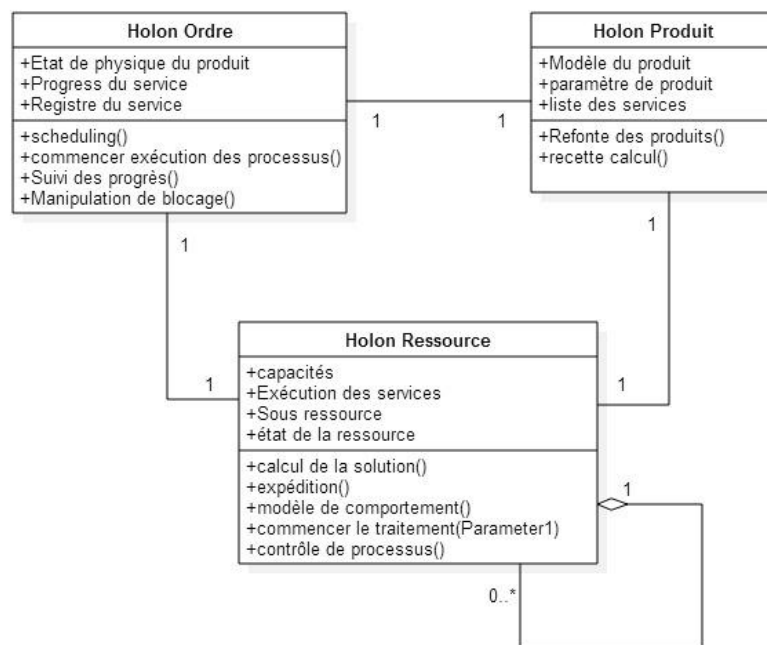


Figure 30. Diagramme de blocs élémentaires de H²CM.

2.2.2 Définition du holon ordre

L'ordre définit l'évolution du système de production. L'ordre doit être négocié à partir d'un objectif de production en utilisant le modèle de produit. Un objectif de production est transformé en un ordre qui spécifie les services et ses paramètres vis-à-vis des services fournis par les ressources atomiques (au sens de l'holarchie), les conditions de qualité à atteindre et les indicateurs permettant de définir lorsque chaque étape de production est terminée.

Le holon Ordre est globalement responsable du planning du système. Il est très représentatif du fonctionnement dual de chaque holon : une partie du holon (noté RT pour temps-réel dans la suite) est dédiée au monitoring du système, tandis que l'autre (noté LA pour Look-Ahead dans la suite) est dédié aux mécanismes de négociation en mode anticipation afin de déterminer le futur planning et les holarchies à créer au sein du système. L'Ordre RT surveille les entrées et sorties du système grâce à des capteurs, afin de vérifier la correspondance avec les besoins exprimés par le Produit RT. Il est aussi responsable de l'application du planning des ressources, résultant de négociations précédentes. L'Ordre LA est un acteur des négociations, comme défini dans la section suivante.

2.2.3 Définition du holon produit

Le holon Produit présente une différence principale avec le même holon considéré dans les architectures discrètes. Traditionnellement, celui-ci est dédié à stocker et communiquer les informations sur la gamme des produits à fabriquer. Dans le cas des HCS, les gammes sont déduites d'une gamme maîtresse, appelée Master Recipe, (ANSI/ISA-95, 2010). Cette gamme présente un ensemble d'opérations type qui peuvent être utilisées pour fabriquer un produit à partir d'un produit brut ayant des caractéristiques physiques comprises dans un certain intervalle. Dans le cas inverse, des ajustements peuvent être réalisés dans cette gamme de sorte que la gamme effectivement appliquée peut être radicalement différente de la Master Recipe. Dans H²CM, cette Master Recipe peut être définie comme une gamme générique, i.e. une liste séquentielle d'opérations à appliquer au produit pour passer du produit brut au produit fini, voir Figure 31, à partir de laquelle la gamme réelle sera déduite en fonction des conditions relevées sur le système. Par exemple, selon la qualité du pétrole brut, il peut être nécessaire d'ajouter ou de rajouter des opérations de filtrage au niveau des premières étapes de la gamme.

Dans le contexte HCS, une spécification orientée-services, comme proposée dans (Gamboa Quintanilla et al., 2014), est bien adaptée pour la spécification du produit, où chaque service est défini par un identifiant et un ou plusieurs paramètres de services. En se référant à la Figure 32, un exemple de service est un service de perçage d'un trou et un paramètre de ce service est le diamètre du trou. La distinction faite ici par rapport à la définition introduite par (Gamboa Quintanilla et al., 2014), est que les paramètres et variables définissant le service peuvent être discrets ou continus. Le diamètre du trou peut ainsi être une variable discrète et prendre des valeurs discrètes prédéfinies, ou peut être une variable continue et prendre toute valeur dans un intervalle.

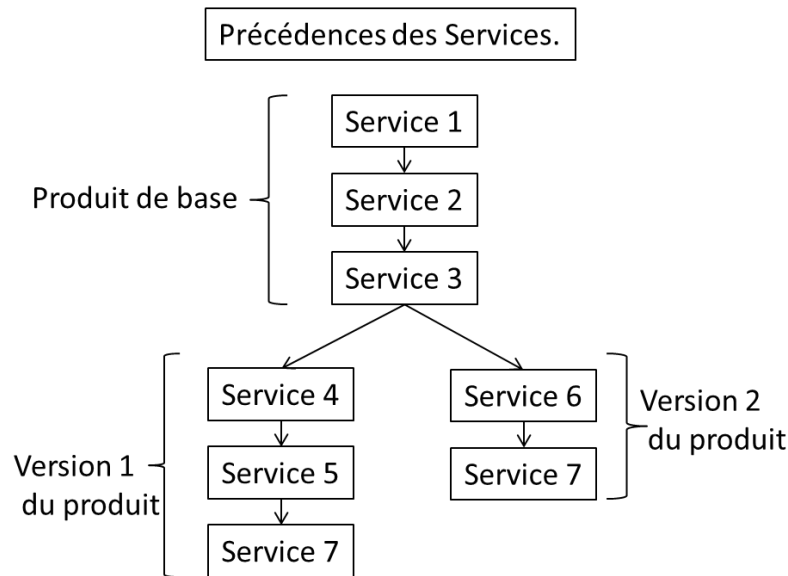


Figure 31. Exemple de liste de services pour fabriquer un produit.

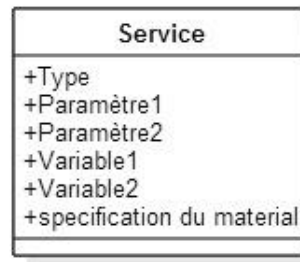


Figure 32. Définition de service.

Dans les systèmes hybrides, l'obtention d'un produit peut avoir besoin de différentes étapes, où les services sont fournis par une ou plusieurs ressources qui forment une holararchie temporaire pour améliorer le produit. Si l'on se réfère à (ANSI/ISA-95, 2010), ceci peut être défini comme la route du produit. Le service rendu par l'holarchie peut lui aussi être composé de plusieurs étapes. Par exemple, le produit doit être chargé, chauffé, et après ça, une réaction chimique se passe qui doit être régulée par une technique quelconque. Cette séquence doit s'exécuter de façon correcte pour obtenir le produit désiré. L'étape correspondant à la charge de la matière première se finira donc lorsque la quantité de produit spécifiée est atteinte. L'étape suivante, qui implique le chauffage du mélange obtenu dans l'étape précédente exige de fournir une quantité d'énergie dépendant de la quantité de produit chargé, et l'alimentation doit être contrôlée par un régulateur en fonction de la même quantité. La fin de cette étape est déterminée par un niveau de température connu dans le modèle du produit. L'étape suivante doit spécifier le régulateur (loi de commande) correspondant à la réaction chimique pour la quantité de matière et les conditions de fin du processus.

Le service va donc être fourni suivant le modèle du produit en déterminant dynamiquement les spécifications pour chaque étape. Lorsque le service est terminé,

l'étape correspondante est finie, et une nouvelle étape doit commencer au moyen d'un autre service qui sera rendu par une autre holarchie de ressources.

La gamme complète est donc distribuée tout au long de l'holarchie, et un holon Produit est spécifié par 3 éléments:

- Indicateurs de qualité du produit brut ;
- Indicateurs de qualité du produit final ;
- Définition du service à exécuter par la ressource.

2.2.4 Définition du holon ressource

Par nature, les HCS sont de grands systèmes, constitués de beaucoup de composants. De ce fait, beaucoup de holons Ressource sont nécessaires pour contrôler le système. Etant donné que la Figure 30 a défini la récursivité des ressources, il est nécessaire de déterminer la plus petite taille d'un holon Ressource. Ces plus petits holons Ressource, appelés ressources atomiques dans la suite de ce document, peuvent être définis comme :

« Une ressource atomique est l'agrégation maximale d'éléments pour laquelle le système d'équations différentielles peut être inversé, dans un délai court relativement à la dynamique du système considéré. »

Concernant les ressources composées, le mécanisme de négociation doit permettre de déterminer la meilleure solution en interrogeant récursivement les ressources composées jusqu'à atteindre les ressources atomiques. Les ressources LA sont parties prenantes de ces mécanismes de négociation. Leur fonction est d'évaluer et transmettre à l'holon Ordre les meilleures valeurs de variables possibles pour obtenir les fonctions et services désirés. Les mécanismes de négociation utilisent un modèle de communication présenté Figure 33, où le langage de négociation des ressources est basé sur la paramétrisation de services, demandés et fournis par des ressources, comme proposé dans (Gamboa Quintanilla et al., 2014). Les holons Ressource RT ont la responsabilité de la commande en ligne des équipements, i.e. le rôle de contrôleur du système présenté précédemment dans la modélisation classique des HCS.

Une autre caractéristique des holons Ressource est leur structure, composée d'une partie physique et d'une partie logique, comme illustré dans la Figure 34. La partie physique est représentée par les équipements de production. La partie logique est une abstraction de la partie physique et contient les modèles de conversion des états continus aux états discrets et vice versa. Les modèles utilisés sont des modèles hybrides qui changent d'état par rapport à des seuils fixés à certaines valeurs des variables continues ou par rapport aux lois de commande utilisées, ce qui change la dynamique du système $x' = f(x,u)$; $u=g_1(x)$. Plusieurs modèles peuvent coexister à l'intérieur du holon Ressource et représenter des caractéristiques telles qu'un modèle de détection de panne, le modèle du comportement souhaité du système ou les modèles inhérents au paradigme holonique tels que les modèles de mécanismes de négociation par exemple.

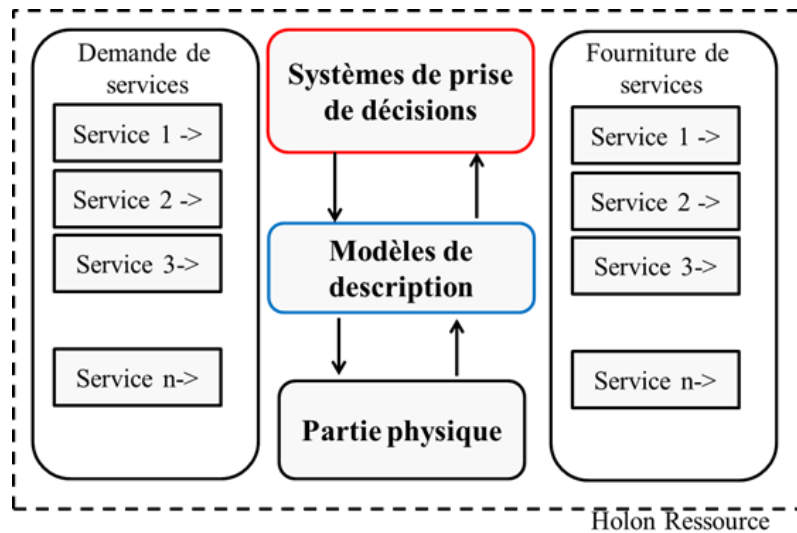


Figure 33. Modèle de communication des holons Ressource.

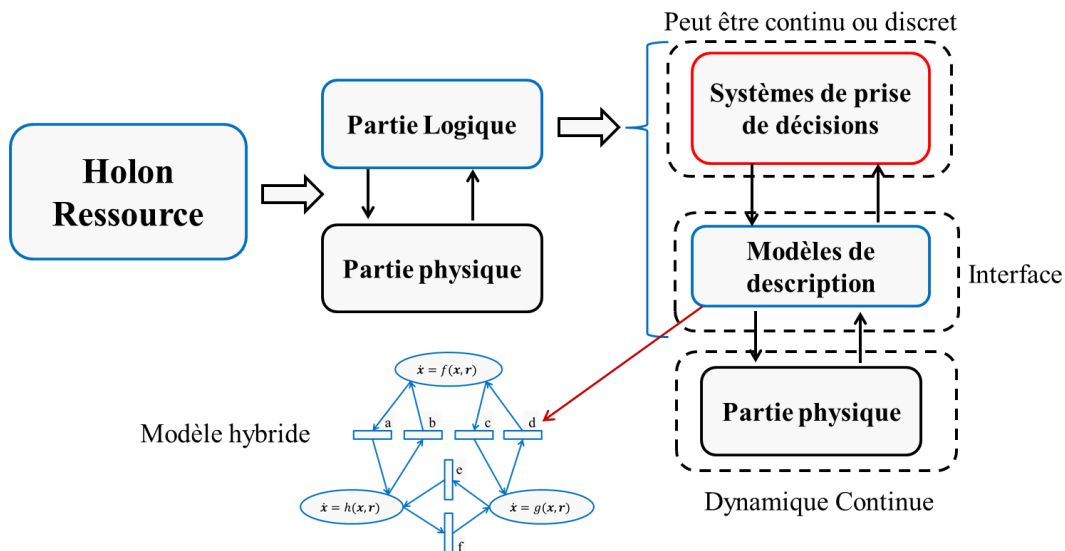


Figure 34. Structure d'un holon Ressource.

2.3 Planification et supervision de la production

Comme expliqué précédemment, l'architecture H²CM reste très similaire aux architectures orientées événements discrets. Cependant, les fonctionnalités de chaque holon sont légèrement différentes, étant donné que certaines tâches sont adaptées du fait de la nature des produits. La Figure 35 propose une comparaison entre l'architecture PROSA de (Van Brussel et al., 1998) et l'architecture H²CM. La première différence pouvant être notée est l'orientation services de l'architecture. A ce niveau de modélisation, cela ne se ressent que par le remplacement des tâches par des services. En ce qui concerne les Ordres, la notion de verrou est supprimée, car ces situations ne sont pas plausibles dans le cas d'une production hybride. Les principales différences peuvent enfin être notées dans les fonctions des Ressources, puisque ces holons ont pris une place plus importante dans l'architecture que classiquement.

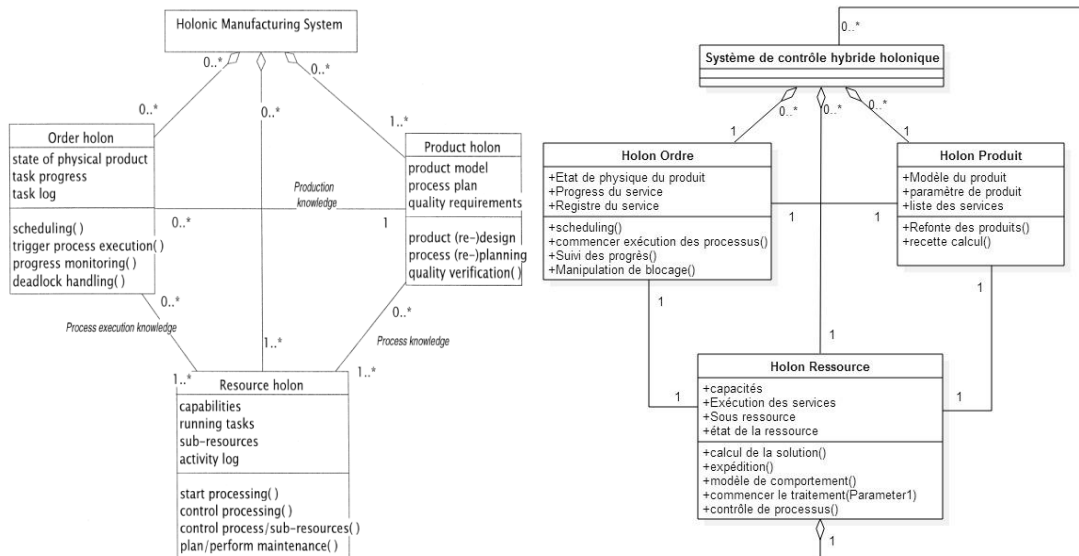


Figure 35. Comparaison entre l'architecture de référence PROSA (gauche) et H²CM (droite).

La Figure 36 propose un diagramme de séquence illustrant les mécanismes de négociation. Celle-ci démarre à l'arrivée d'un nouvel ordre au niveau d'agrégation N+1. L'objectif est d'illustrer les interactions avec les ressources du niveau N, alternativement ressources atomiques ou elles-mêmes composées. Cette séquence est exécutée chaque fois qu'une nouvelle négociation est déclenchée (fonctionnalité LA), i.e. lorsqu'un nouvel ordre doit effectivement entrer dans le système et doit être inséré dans le planning des ressources, ou lorsqu'un ordre doit être réévalué dans le cas d'une impossibilité de réalisation du résultat de la précédente négociation au coût fixé précédemment.

La séquence n°12 (notée Dispatch()) est la base du comportement récursif des ressources. Dans le contexte de ressources composées, cette séquence répète l'ensemble des séquences n°7 à 17, jusqu'à atteindre le niveau des ressources atomiques. De ce fait, la Figure 36 peut être lue comme 2 fonctions Dispatch() imbriquées, du niveau N+1 au niveau N. La séquence n°13 permet aux ressources de calculer la meilleure solution parmi les différentes propositions offertes par les sous-ressources. La séquence n°14 consiste généralement en la résolution d'équations différentielles au sein d'une ressource atomique. La séquence n°3 consiste en la détermination de la gamme relative au produit faisable sur le système dans le laps de temps identifié, prenant en compte les capacités des équipements mais sans connaître le planning de chaque ressource.

2.4 Supervision du système

La supervision est réalisée par l'holon Ordre RT associé au holon Ressource dans la structure récursive proposée dans H²CM. En conséquence, la supervision est directe entre les ordres et les ressources. La structure du superviseur est relativement simple et fixe, centrée sur l'accomplissement du planning suivant la gamme du produit associé. Le superviseur détecte des événements connus à partir du modèle du produit pour déterminer lorsque l'évolution arrive à un nouveau pas.

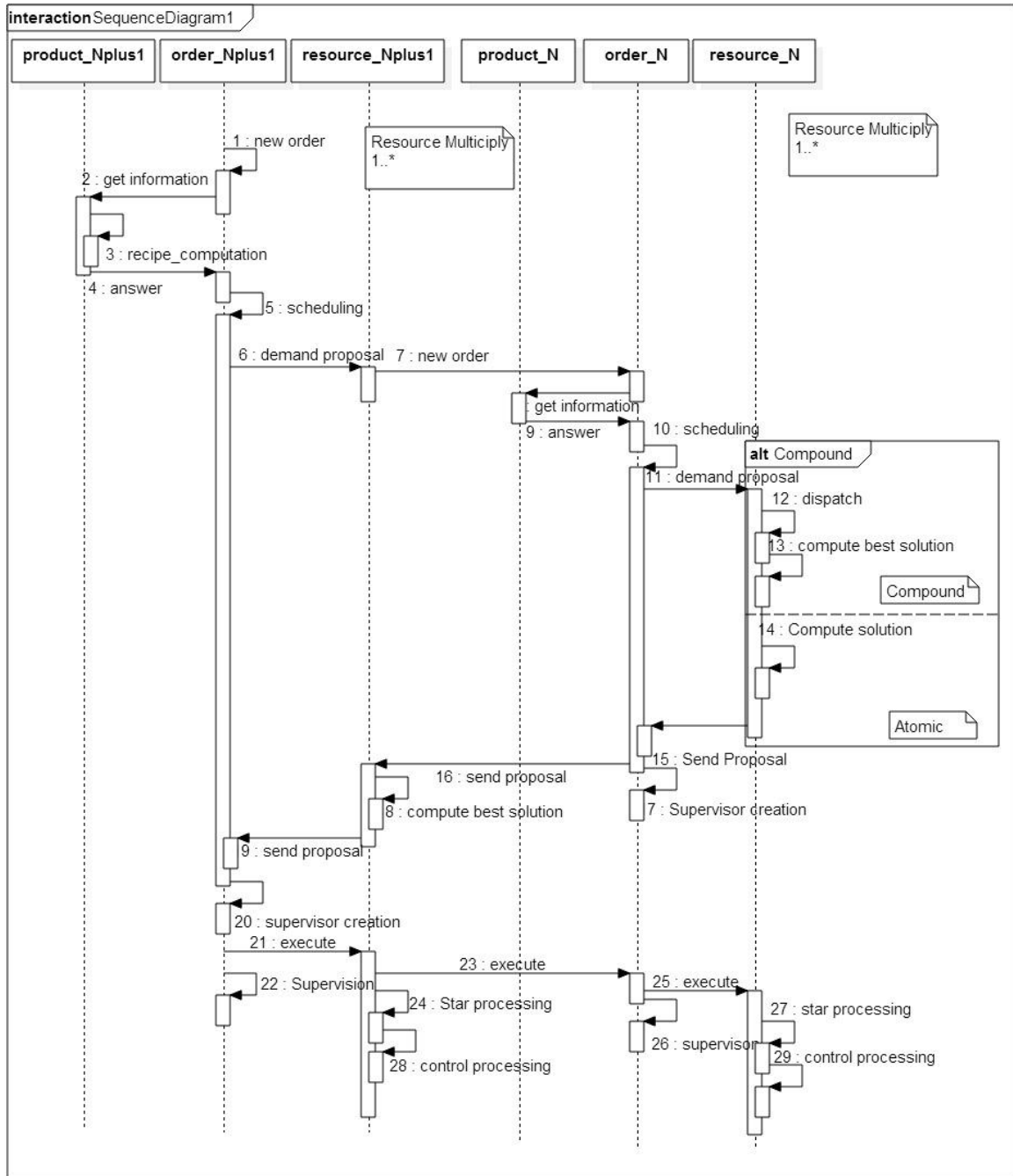
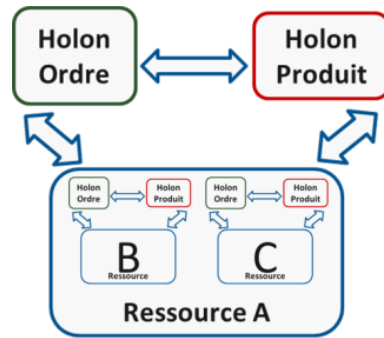


Figure 36. Diagramme de séquence des mécanismes de négociation.

Le superviseur décide alors le changement du régulateur pour le nouveau mode opératoire défini a priori pour la ressource. Une autre tâche du superviseur est de vérifier l'état du holon Ressource. Pour illustrer cette tâche, deux scénarios sont présentés. Dans le premier, l'holon Ordre est associé à une ressource atomique. Dans ce cas, l'état du superviseur ne dépend que de l'état de la ressource atomique. Dans le second scénario l'holon Ordre est associé à une ressource composée. Dans ce cas, l'état du superviseur dépend de la composition des états des superviseurs des ressources composant ce holon Ressource. Les états de ressources définis dans ces travaux sont « Ressource en défaillance », « Ressource en maintenance », « Ressource opérationnelle » et « Ressource

en mode dégradé ». Si les états des superviseurs sont représentés par des automates, alors l'état du superviseur composé est un automate issu de la composition de ces automates. La Figure 37 illustre un exemple d'une ressource A composée de deux ressources redondantes B et C. Selon les états des superviseurs des ressources B et C, l'état du superviseur de la ressource A est exprimé. Ce cas est représentatif de ressources positionnées en parallèle. Le cas de ressources placées en séries est trivial, car l'état de la ressource composée correspond à l'état du superviseur le plus limitatif.



Resource B	Resource C	Resource A
Opérante	Opérante	Opérante
Opérante	Maintenance	Opérante - dégradée
Opérante	Défaillante	Opérante - dégradée
Opérante	Opérante - dégradée	Opérante - dégradée
Maintenance	Maintenance	Maintenance
Maintenance	Défaillante	Défaillante
Maintenance	Opérante - degraded	Opérante - dégradée
Défaillante	Défaillante	Défaillante
Défaillante	Opérante - dégradée	Opérante - dégradée
Opérante - dégradée	Opérante - dégradée	Opérante - dégradée

Figure 37. Expression des états du superviseur d'une ressource composée.

2.5 Revue du chapitre

Ce chapitre passe dans un premier temps en revue quelques travaux sur l'utilisation d'agents et de l'architecture holonique pour piloter des systèmes continus ou hybrides, en soulignant les caractéristiques les plus pertinentes de ces solutions. Puis, nous avons proposé un tableau comparatif de ces travaux.

Ensuite, nous avons proposé un modèle de Contrôle hybride holonique où sont définis la structure des holons qui les constituent, et les mécanismes de planification et de supervision. Dans le chapitre suivant, un exemple de mise en œuvre du modèle proposé est présenté à l'aide d'un cas classique de la littérature appelé système de commutation d'arrivées.

Chapitre 3

3 Cas d'étude de systèmes de commutation d'arrivée

Le chapitre précédent a présenté notre contribution principale, une architecture générique de modélisation holonique de systèmes hybrides. L'objectif de ce chapitre est d'appliquer cette architecture sur un cas d'étude classique de la littérature et de comparer ses performances par rapport à une solution optimale calculée offline grâce à un algorithme d'ordonnancement. Le cas d'étude choisi est relativement simple, mais a déjà une taille relativement critique pour la modélisation MILP présentée ci-après.

Le but de cette comparaison est de valider les performances de l'architecture proposée sur un cas d'étude où la solution optimale est calculable avant d'étendre dans le chapitre suivant notre proposition à un cas d'étude beaucoup plus complexe.

Dans ce chapitre, une description du système de commutation d'arrivée est tout d'abord présentée, suivie de l'explication du modèle d'ordonnancement optimal du nombre de changements de serveur. Une modélisation du système, un ensemble d'expérimentations et les résultats associés sont ensuite proposés, avant une étude plus spécifique sur la stabilité du système de contrôle obtenu.

3.1 Description du système de commutation d'arrivée

Cette section décrit un exemple de système hybride, appelé système de commutations d'arrivée (de flux), et présenté initialement dans (Chase et al., 1993).

Le système est constitué de n réservoirs et un serveur. Le fluide est retiré de chacun des réservoirs i à un taux fixe $\rho_i > 0$. Pour compenser, le serveur fournit du fluide à un réservoir sélectionné au taux de v . Nous supposons que le système est fermé, de façon que $\sum_{i=1}^n \rho_i = v$. L'emplacement du serveur est une variable de commande, et peut être sélectionné en utilisant une politique de réalimentation. Le déplacement du serveur modifie la topologie du flux, et par conséquent permet le contrôle du niveau du réservoir. La Figure 38 montre le système avec $n = 3$ et le serveur localisé sur le réservoir 1.

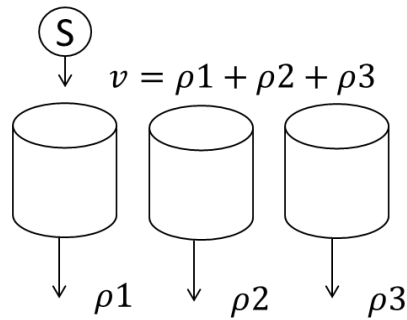


Figure 38. Système de commutations d'arrivée avec $n=3$.

La politique de contrôle proposée est une politique de seuil de la forme suivante : « Attribuer un seuil pour chaque réservoir, et instantanément déplacer le serveur vers un réservoir dans lequel le niveau est inférieur au seuil assigné ». Il est à noter que l'emplacement du serveur est sélectionné sur la base d'une observation quantifiée de l'état du réservoir, et le mouvement du serveur est déclenché par un « événement discret » (par exemple quand un réservoir se vide). Dans la version la plus simple de ce dispositif, on prend tous les seuils égaux à zéro, et on passe le serveur d'arrivée à chaque fois qu'un réservoir est vide.

Plusieurs auteurs ont utilisé ce système comme illustration de travaux divers, notamment sur la détection de défaillance, car il a la particularité d'avoir un comportement chaotique lié aux conditions initiales. Dans notre cas il a été choisi pour sa simplicité et la possibilité de reconfigurer simplement le système, notamment en modifiant le nombre de réservoirs considérés. Ceci permettra de qualifier la flexibilité du système de contrôle proposé.

3.1.1 Notations utilisées

Pour l'ensemble de ce chapitre, les notations suivantes sont définies:

- T_i : réservoir $n^o i$;
- N_t : Nombre de réservoirs;
- $S_i=(T_i, \text{date})$: réservation du serveur par un réservoir à un temps dénommé « date »;
- P : Planning, i.e. ensemble des réservations. En corollaire, $P(\text{date})$ désigne le réservoir vers lequel le serveur doit se déplacer au temps date ;
- t_p : date actuelle;
- T_s : réservoir sur lequel est actuellement le serveur ;
- $x_i(t)$: quantité de liquide dans un réservoir i au temps $t > 0$;
- $x(t) = (x_1(t), \dots, x_n(t))$: état des réservoirs à l'instant t .

3.2 Modélisation du système de commutation d'arrivée

A $t=0$, nous supposons que $x_i(0) \geq 0$ avec $\sum_{i=1}^n x_i(0) = h$. Si à l'instant t le serveur se trouve en position j , alors le serveur restera à j jusqu'à ce qu'un autre réservoir soit vide. Cet événement aura lieu après un temps $\tau = \min_{i \neq j} \{x_i(t)/\rho_i\}$. Pour $t \leq s \leq t + \tau$ l'état des réservoirs est déterminé par l'ensemble d'équations linéaires :

$$x_i(s) = \begin{cases} x_i(t) - \rho_i(s - t), & \text{si } i \neq j; \\ x_i(t) + (v - \rho_i)(s - t), & \text{si } i = j; \end{cases} \quad (1)$$

Ainsi, l'idée du contrôle est de fournir du fluide dans les réservoirs de manière à ce qu'aucun réservoir ne déborde et d'éviter que deux ou plusieurs réservoirs soient vides en même temps. Une analogie peut être faite entre le système de commutations d'arrivée et un système de distribution d'eau où le serveur est représenté par le château d'eau et les réservoirs sont représentés par les réservoirs dans les quartiers. L'objectif est qu'aucun réservoir de quartier ne soit vide ni ne déborde. De plus, s'il y a deux réservoirs de quartier vides, alors il y aura un problème de manque d'eau au niveau du quartier dont le réservoir ne sera pas rempli par le serveur.

Pour illustrer ce qu'il se passe avec la condition (1), considérons le système avec $N=3$ et un seuil égal à zéro (Figure 39(a)). Soit $X = \{x: \sum_i x_i = h, x_i \geq 0, \text{ et pour un } j, x_j = 0\}$ Dans ce cas, Toutes les trajectoires possibles de $x_i(t)$ peuvent être visualisées géométriquement dans un triangle équilatéral ayant pour côtés X_1, X_2, X_3 et pour sommets $v_1=(h,0,0)$, $v_2=(0,h,0)$ et $v_3=(0,0,h)$. Les sommets représentent les états pour lesquels deux réservoirs sont vides simultanément.

Si les seuils sont définis par des valeurs $x_i^{min}(t) > 0$ pour tous les réservoirs, la région représentée par toutes les trajectoires possibles de $x_i(t)$ change, comme indiqué sur la Figure 39(b), et la valeur de $\tau = \min_{i \neq j} \left\{ \frac{x_i(t) - x_i^{min}(t)}{\rho_i} \right\}$

Si des seuils haut et bas sont définis simultanément, $x_i^{max}(t) > x_i(t) > x_i^{min} > 0$ pour tous les réservoirs, la région représentée par toutes les trajectoires possibles de $x_i(t)$ change, comme indiqué sur la Figure 39(c), et la valeur de τ est:

$$\tau = \min \left\{ \begin{cases} \left\{ \frac{x_i(t) - x_i^{min}(t)}{\rho_i} \right\} & \text{if } i \neq j \\ \left\{ \frac{x_i^{max}(t) - x_i(t)}{h - \rho_i} \right\} & \text{if } i = j \end{cases} \right.$$

Quand deux réservoirs sont vides en même temps, le système devient non déterministe et le contrôle classique ne donne aucune solution. Pour trouver une solution, les systèmes hybrides utilisent le modèle du système complet. Le contrôle utilise le modèle d'ensemble de réservoirs (Labinaz et al., 1997) (Ushio et al., 1995) (Labinaz et al., 1996), qui est un modèle complexe, rigide et difficile à modifier. Ainsi, si un réservoir est ajouté, le modèle de tout le système doit être changé pour reconfigurer le contrôle, caractéristique qui montre la rigidité du système classique. Enfin ce modèle est fortement dépendant des

conditions initiales (Chase et al., 1993)(Tian, 2005). Ces travaux montrent que, pour certaines conditions initiales, le contrôle hybride « classique » ne fonctionne pas, et propose donc de limiter l'espace de conditions initiales admissibles.

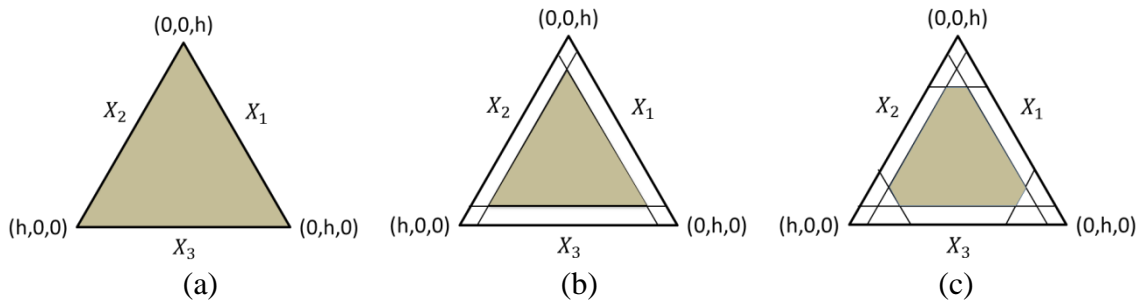


Figure 39.(a) Région représentée par toutes les trajectoires possibles de $x_i(t)$, (b) avec bornes inférieures non-nulles, (c) avec bornes inférieures et supérieures.

3.2.1 Programmation JAVA

La simulation du système de commutation d'arrivée a été réalisée à l'aide d'un programme Java. Il a été créé quatre classes (Figure 40) :

- la classe de ressources, qui est la classe qui communique avec la classe ordre et la classe produit du modèle de contrôle proposé ;
- la classe SimDuServer, spécialisation de ressource, abstraction du serveur, qui utilise la position du serveur pour générer des événements qui sont utilisés dans le système de commande de commutation du serveur ;
- la classe SimDuReservoir, spécialisation de ressource, abstraction du réservoir, qui utilise les données des niveaux des réservoirs pour générer des événements qui sont utilisés dans le système de commande de commutation du serveur ;
- et la classe SystemesDesReservoirEtServeur, spécialisation de ressource, qui contient toutes les équations mathématiques présentées dans la section précédente. Cette classe est utilisée pour représenter le comportement du système réel générant l'évolution des valeurs des variables du niveau des réservoirs, du débit du réservoir, la position actuelle du serveur et le débit du serveur.

3.2.2 Fonction objectif du pilotage

La fonction objectif retenue est de minimiser le nombre de changements de position du serveur (switches) tout en respectant les contraintes de niveaux minimum et maximum de chaque réservoir.

La section suivante introduit un algorithme de calcul offline de la solution optimale minimisant le nombre de switches. Cet algorithme a été imaginé en faisant un parallèle entre ce cas d'étude et un problème de minimisation du nombre de réglages d'une « single machine », représentant ici le serveur. Les contraintes de niveaux sont inspirées des fenêtres d'exécution typiques d'un problème type HSP – Hoist Scheduling Problem (Che et al., 2011).

La solution optimale nous servira de borne inférieure afin de comparer les résultats donnés par la modélisation holonique du système.

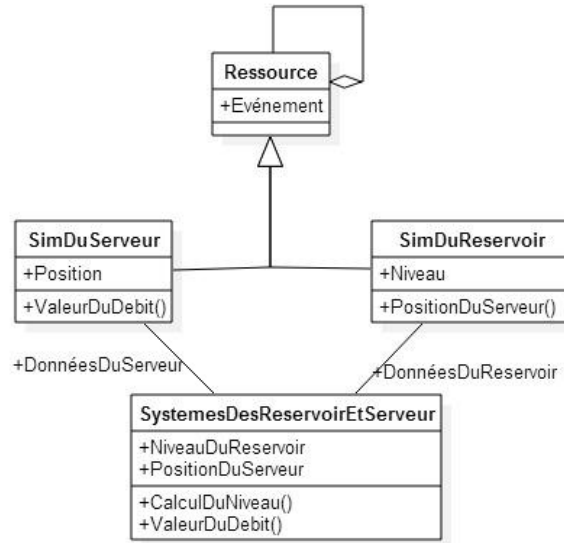


Figure 40. Diagramme de classes du système des réservoirs.

3.3 Modèle d'ordonnancement optimal du nombre de changements de serveur

Cette section est dédiée à la description de la structure d'un modèle MILP (Mixed Integer Linear Programming) destiné à calculer la solution optimale du nombre de switches. L'ensemble des variables considérées est le suivant :

- $L_{v_{i,c}}$ représente le niveau du réservoir i au début d'une phase c ;
- D_c représente la durée d'une phase c ;
- $Q_{i,c}$ représente la quantité d'eau ajoutée à un réservoir i durant une phase c ;
- $O_{i,c}$ représente la quantité d'eau perdue par un réservoir i durant une phase c ;
- $R_{i,c}$ est un booléen, égal à 1 si le serveur est positionné sur le réservoir i durant la phase c ;
- $Sw_{i,j,c}$ est un booléen, égal à 1 si le serveur se déplace du réservoir i au réservoir j au début de la phase c .

La formulation du problème peut donc être écrite par la fonction objectif :

$$\text{Min } Z = \sum_i \sum_j \sum_c Sw_{i,j,c} \quad (0)$$

Et l'expression de l'ensemble des contraintes :

$$\forall i, \forall c, Lv_{i,c} \leq \text{LevelMax} \quad (1)$$

$$\forall i, \forall c, Lv_{i,c} \geq \text{LevelMin} \quad (2)$$

$$\forall i, \forall c, Lv_{i,c} = Lv_{i,c-1} + Q_{i,c-1} - O_{i,c-1} \quad (3)$$

$$\forall i, \forall c, Q_{i,c} = \begin{cases} D_c * Q, & \text{if } R_{i,c} = 1; \\ 0, & \text{otherwise;} \end{cases} \quad (3')$$

$$\forall i, \forall c, O_{i,c} = O_c * D_c \quad (3'')$$

$$\forall c, \sum_i R_{i,c} = 1 \quad (4)$$

$$\forall i \neq j, \forall c, R_{i,c-1} + R_{j,c} - Sw_{i,j,c} \leq 1 \quad (5)$$

L'équation (0) exprime la minimisation du nombre de switches. L'ensemble de contraintes (1) et (2) assurent que le niveau des réservoirs reste entre les limites définies. La contrainte (3) permet d'assurer la mise à jour du niveau des réservoirs à chaque phase en prenant en compte à la fois la quantité d'eau ajoutée si le serveur est positionné sur le réservoir durant la phase c (3') et le débit d'eau perdue par le réservoir durant cette même phase (3''). La contrainte (4) exprime que, pour chaque phase, le serveur ne peut pas remplir ni plus ni moins d'un réservoir à la fois. Enfin, la contrainte (5) assure le lien entre switches et remplissages. Des contraintes additionnelles ont été nécessaires pour linéariser les expressions, lier les variables et renforcer le modèle.

3.4 Holonic Hybrid Control model sur des systèmes de commutation d'arrivée.

Le système a été modélisé conformément à la représentation donnée en Figure 41. Les holons Ordre sont en charge du respect des débits de sortie des ressources exprimés dans le holon Produit. Pour assurer ces conditions, deux fonctions doivent être implémentées.

Tout d'abord, en mode prospectif, le holon négocie avec chaque ressource Réservoir et avec la ressource Serveur afin d'établir le planning du serveur. Ce processus est décrit dans l'algorithme de la Figure 42. L'idée principale est d'essayer de maximiser la durée pendant laquelle le serveur reste dans la même position, tout en évitant la situation dans laquelle deux réservoirs seraient vides en même temps (voir Figure 43 (a)). De ce fait, les réservoirs essaient d'anticiper la réservation du serveur si la date à laquelle ils seront vides est déjà réservée par un autre réservoir (voir Figure 43 (b) et (c)). Une durée de réservation est aussi définie (timelapse). Ce timelapse représente la durée initiale pendant laquelle le réservoir essaie de réserver le serveur. Si aucune solution ne peut être trouvée par le système avec ce timelapse initialement défini, alors l'algorithme décrémente cette durée et essaie de trouver un créneau de réservation. Si aucune solution ne peut être trouvée avec un timelapse supérieur ou égal à 1 pas de simulation (prédéfini lors de la discrétisation du temps), alors le planning est intégralement vidé et l'algorithme est de nouveau exécuté pour planifier l'ensemble des réservoirs.

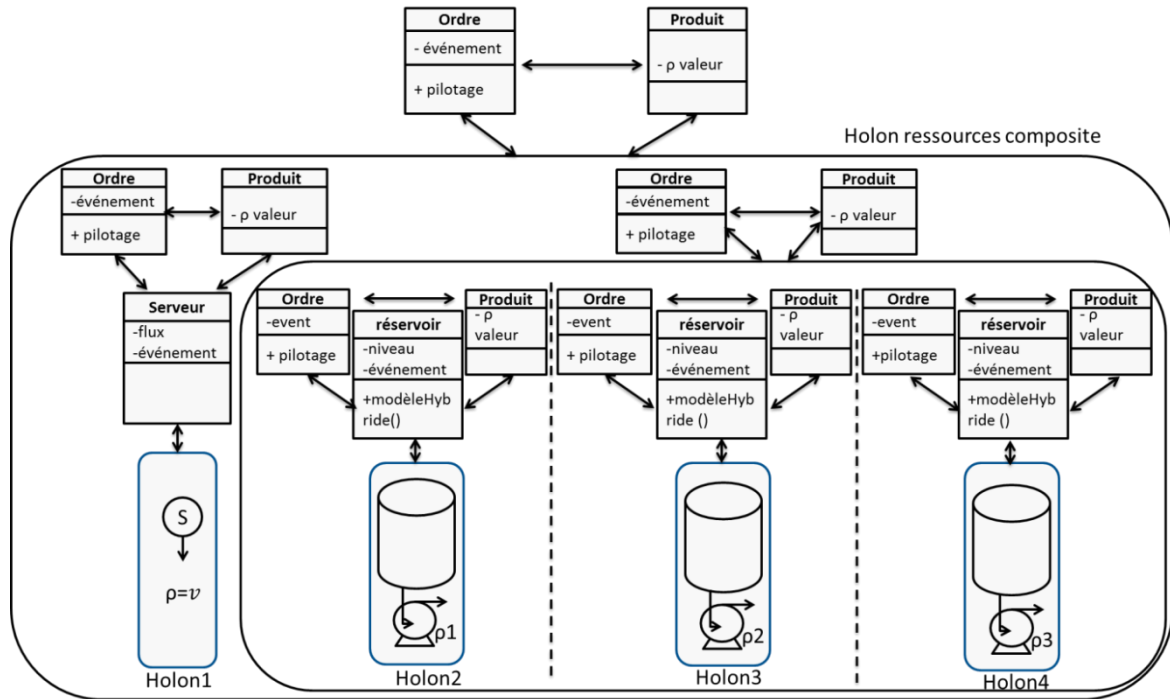


Figure 41. Modélisation holonique du système

La seconde fonction est exécutée en temps réel. Le holon déclenche le changement de position du serveur, suivant l'algorithme présenté Figure 44. À chaque fois que le serveur change de position, les réservations du nouveau réservoir servi sont retirées du planning et le réservoir qui n'a plus le serveur lance une procédure de réservation (Figure 42). Il ne s'agit donc pas d'une optimisation globale, mais bien d'un ordonnancement en ligne, puisque les réservations déjà réalisées ne peuvent pas être remises en cause.

3.5 Flexibilité de la commande

Tout comme son équivalent la robustesse (Cardin et al., 2013), la flexibilité est une grandeur difficile à quantifier. Tout au plus pouvons-nous comparer plusieurs configurations et tirer des enseignements sur une évolution de la flexibilité relative entre les cas considérés.

Pour cette étude, une comparaison avec le contrôle classique de (Chase et al., 1993) a été choisie. Ce contrôle n'est probablement pas le plus performant de la littérature, mais permettait d'avoir une base de comparaison avec des temps de calculs similaires à l'approche présentée ici et sur le même cas étudié. Tout d'abord, la question était d'évaluer le manque de flexibilité de ce contrôle. En effet, celui-ci est réputé pour ne pas pouvoir donner de solution satisfaisante pour un certain nombre de configurations de niveaux de réservoirs. Pour cela, un certain nombre d'instances aléatoires ont été tirées et simulées en utilisant la méthodologie proposée par (Chase et al., 1993). Lorsque la simulation se trouve dans une configuration où la prochaine position du serveur ne peut pas être définie de manière déterministe, l'instance est considérée comme échouée.

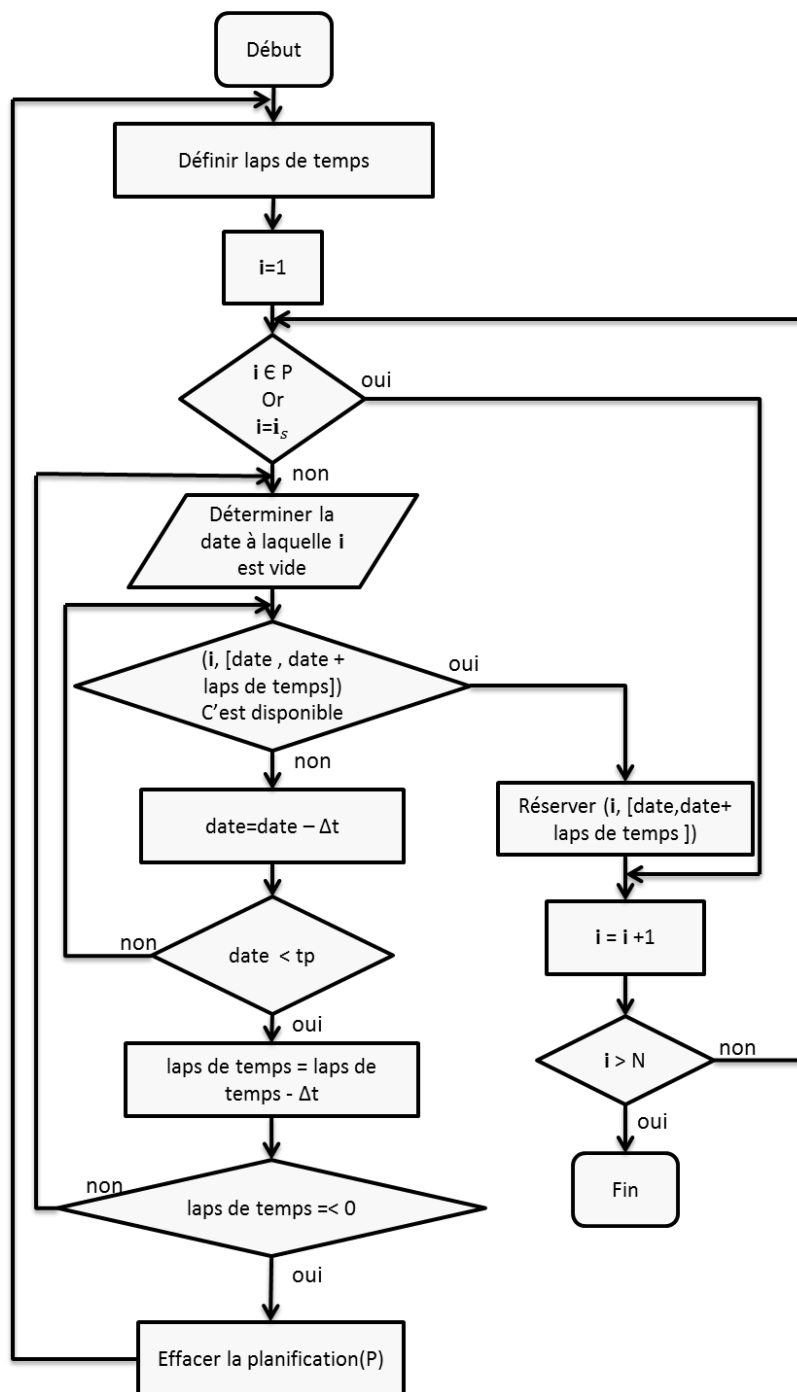


Figure 42. Algorithme de définition du planning de réservations de la position du serveur

De plus amples détails sur le protocole expérimental seront donnés dans la section 3.7 qui reprend cette expérimentation dans un cadre plus large. La Figure 45 montre que le pourcentage d'instances échouées se stabilise en fonction de l'horizon de simulation considéré entre 45 et 50%. L'extrapolation qui peut être intuitivement faite à partir de ce résultat est que de nombreuses configurations issues de la reconfiguration (enlever ou rajouter des réservoirs en temps réel) ne permettront pas de trouver des solutions satisfaisantes.

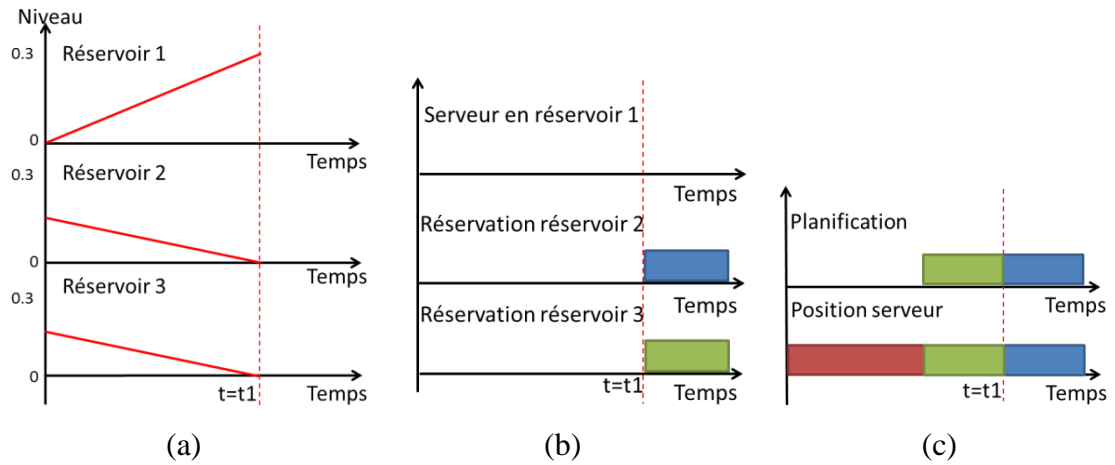


Figure 43 (a) Comportement du niveau des réservoirs, deux réservoirs vident au même temps. (b) Réservation des réservoirs, (c) Planification et positionnement du serveur.

Pour chaque pas de temps, Δt

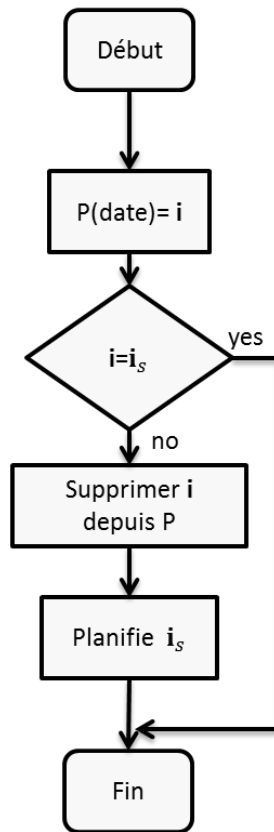


Figure 44. Algorithme de supervision de la position du serveur

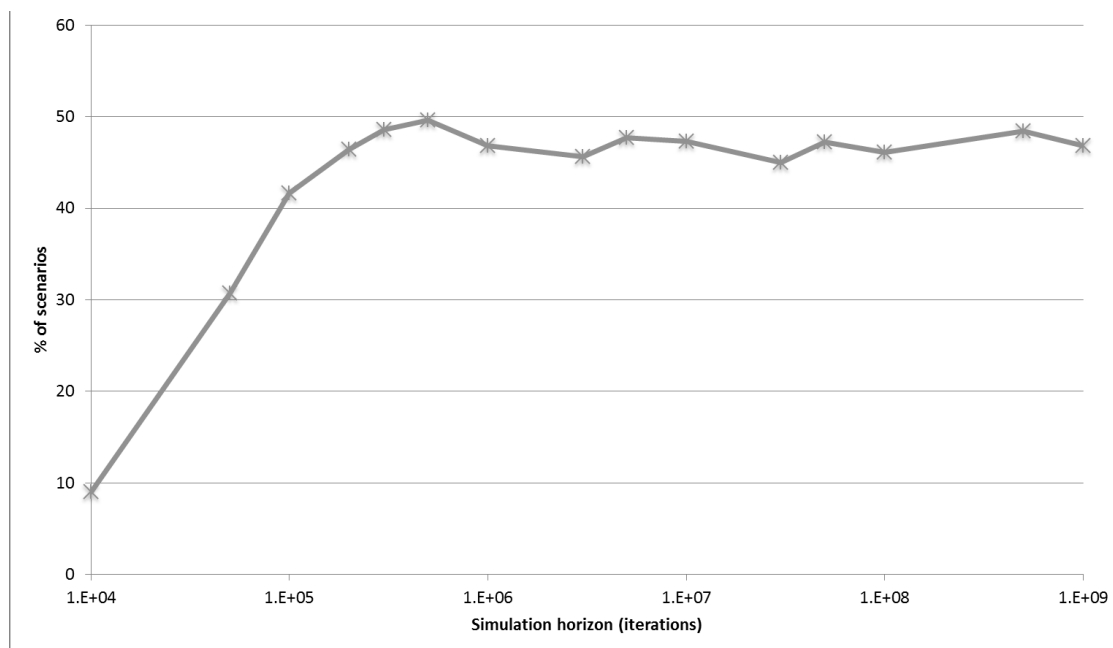


Figure 45. Evolution de la proportion d'instances inadaptées au contrôle classique de (Chase et al., 1993) en fonction de l'horizon de simulation

De la même manière, une expérimentation a été réalisée avec le contrôle holonique proposé dans ce document. L'idée est de faire évoluer le nombre de réservoirs en temps réel lors des simulations. La Figure 46 illustre le cas où le système démarre avec 3 réservoirs, puis à la 600^e itération, sept nouveaux réservoirs sont ajoutés. Lorsque ces réservoirs sont ajoutés, pour satisfaire aux conditions explicitées auparavant, on peut noter que la pente positive, représentant le débit de remplissage du serveur, augmente à partir de la 600^e itération, puisque le débit est incrémenté du débit de fuite de chacun des réservoirs ajoutés afin de satisfaire à la conservation de la quantité d'eau dans les réservoirs. Dans le cas général comme dans celui-ci en particulier, le contrôle holonique a toujours trouvé une solution en temps réel.

La Figure 47 illustre l'évolution de la position du serveur en fonction du nombre d'itérations. Bien évidemment, on peut noter que l'augmentation du nombre de réservoirs entraîne une augmentation du nombre de changements de position du serveur. Ce nombre de changements sera spécifiquement étudié dans la section suivante.

3.6 Optimalité de la commande

3.6.1 Définition des instances

Les réservoirs sont supposés avoir le même volume, normalisé à 1 pour des raisons de simplification. Le débit de sortie est exprimé en pourcentage du volume par pas de calcul. Ainsi, un réservoir initialement plein à 10% (niveau initial=0.1) sera vide après 100 pas de calcul si le débit de sortie est égal à 0.1. Le débit du serveur est, par définition, la somme des débits de sortie de tous les réservoirs.

La Table 3 présente les paramètres définissant les différents scénarios étudiés dans ces expérimentations. Chaque instance est définie par le niveau initial et le débit de sortie de chaque réservoir.

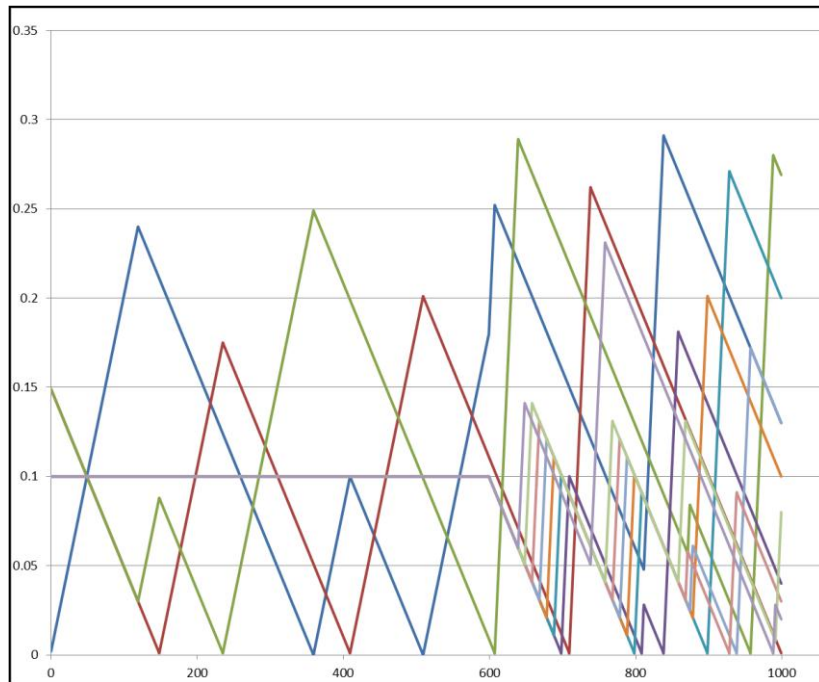


Figure 46. Evolution des niveaux de 3 puis 10 réservoirs en fonctions du nombre d'itérations

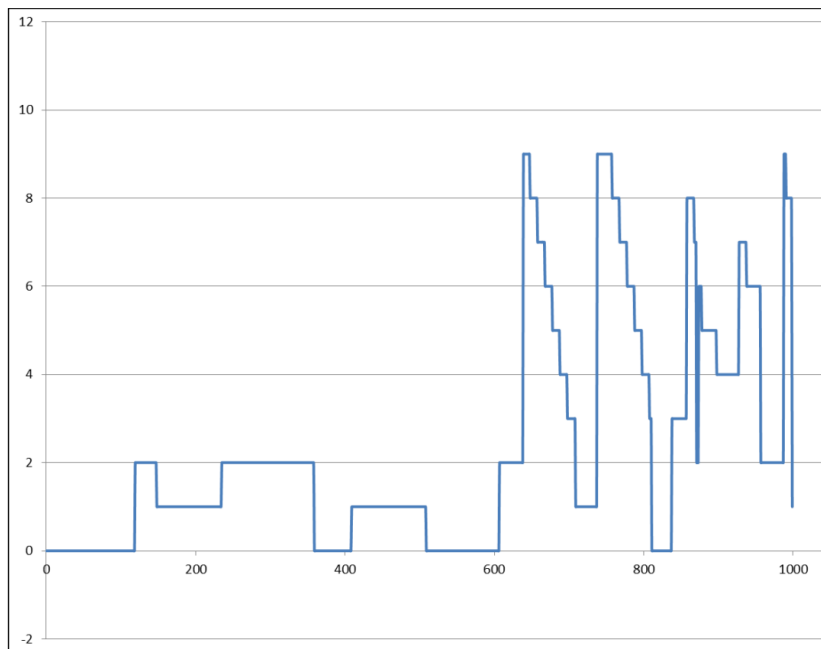


Figure 47. Position du serveur en fonction du nombre d'itérations pour 3 puis 10 réservoirs

Les scénarios 1 et 2 ont été choisis afin que deux réservoirs soient vides en même temps avec différentes conditions initiales. L'objectif de ces scénarios est d'observer le

comportement de la commande vis-à-vis de cette situation particulière, généralement bannie du contrôle hybride classique, comme expliqué précédemment. Les scénarios 3 et 4 ont été conçus avec des conditions extrêmes sur les niveaux initiaux et les débits de sortie. La différence se fait sur la position du fort débit de sortie, situé soit sur un réservoir initialement vide ou initialement plein.

Table 3. Scénarios étudiés

Scénario	Réservoir	1	2	3
1	Niveau initial	0	0.15	0.15
	Débit de sortie	0.1	0.1	0.1
2	Niveau initial	0	0.135	0.165
	Débit de sortie	0.1	0.09	0.11
3	Niveau initial	0	0.29	0.01
	Débit de sortie	0.28	0.01	0.01
4	Niveau initial	0	0.29	0.01
	Débit de sortie	0.01	0.28	0.01

3.6.2 Expérimentations

Pour chaque méthode, un paramètre a été identifié comme influençant la qualité des résultats:

- Les temps de calcul de l'algorithme MILP sont influencés par la borne supérieure du nombre de switches donné a priori au début du calcul : plus la borne est haute, plus les temps de calcul sont longs. De ce fait, à titre de comparaison, deux jeux de calcul ont systématiquement été lancés, l'un avec une borne supérieure très haute (cas appelé « sans borne supérieure » dans la suite de ce document) et l'autre avec le nombre de switches minimal donné par la commande holonique.
- Pour la commande holonique, le timelapse initial influence le nombre de switches obtenu. Si le timelapse est trop court, la commande tend à être instable, changeant le serveur de position à presque chaque pas de temps. Si le timelapse est trop long, le réservoir planifié en premier contraindra tellement le planning qu'il empêchera le système de correctement anticiper les éventuels problèmes.

Le programme JAVA présenté précédemment a été lancé 104 fois sur chaque instance:

1. 50 simulations sur 1000 pas de temps en utilisant la commande holonique, avec un timelapse initial augmentant de 1 à 50 ;

2. 50 simulations sur 2000 pas de temps en utilisant la commande holonique, avec un timelapse initial augmentant de 1 à 50 ;
3. Calcul sur 1000 pas de temps de la solution optimale en utilisant l'algorithme MILP sans borne supérieure prédéfinie;
4. Calcul sur 2000 pas de temps de la solution optimale en utilisant l'algorithme MILP sans borne supérieure prédéfinie;
5. Calcul sur 1000 pas de temps de la solution optimale en utilisant l'algorithme MILP en utilisant la meilleure solution trouvée avec la commande holonique comme borne supérieure;
6. Calcul sur 2000 pas de temps de la solution optimale en utilisant l'algorithme MILP en utilisant la meilleure solution trouvée avec la commande holonique comme borne supérieure.

Les temps de calcul de l'algorithme MILP seront les seuls à être présentés, car la durée nécessaire au programme pour simuler le comportement du système en appliquant la commande holonique pour de si simples instances est toujours inférieure à 1ms.

3.6.3 Résultats et discussion

Les Figure 48, Figure 49, Figure 50 et Figure 51 illustrent l'évolution des niveaux de réservoirs obtenus par le contrôle holonique sur un horizon d'environ 2000 itérations (3000 pour le scénario 3 pour avoir une vision plus globale du comportement).

La Figure 52 montre l'évolution de la performance de la commande holonique selon le timelapse initial choisi sur 2000 pas de temps du scénario 2. L'influence attendue de ce paramètre est confirmée, mais pas sur tous les scénarios (Scénario 3 par exemple donne toujours les mêmes résultats de simulation).

La Table 4 présente l'ensemble des résultats obtenus. Etant donné que les résultats de simulation évoluent avec le timelapse initial, il a été choisi de n'indiquer que les timelapses initiaux pour lesquels les résultats étaient les meilleurs, puis quel pourcentage de timelapses donnent les résultats optimaux, une dégradation de performance de 0 à 20, 40, 60 et plus de 60% par rapport à l'optimal calculé par le MILP.

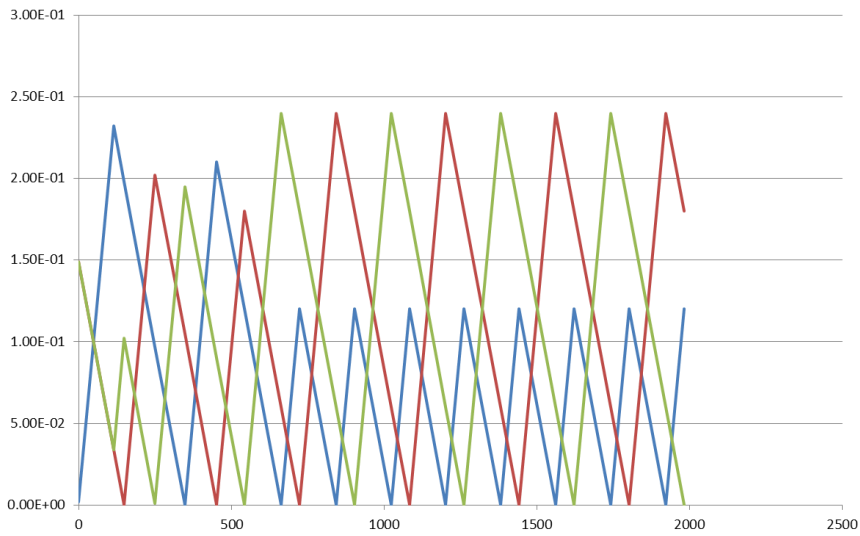


Figure 48. Scénario 1, 2000 itérations, timelapse de 34

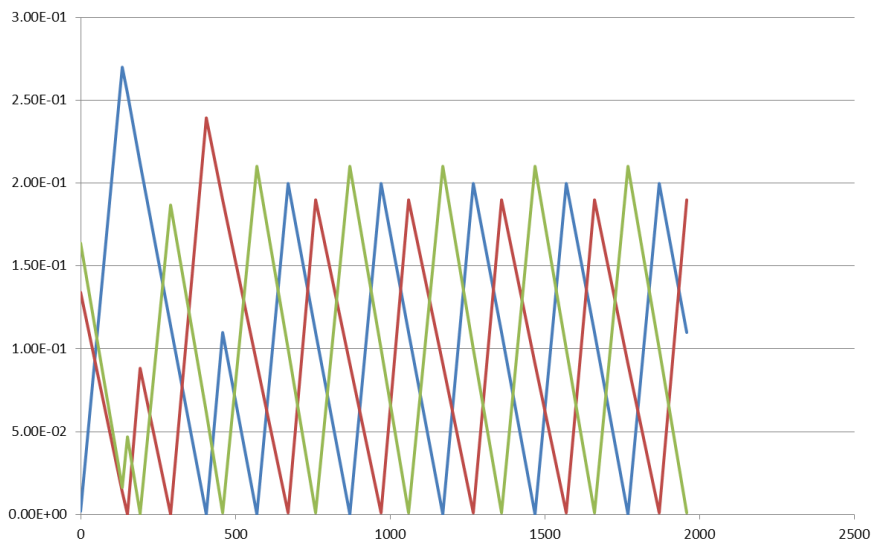


Figure 49. Scénario 2, 2000 itérations, timelapse de 16

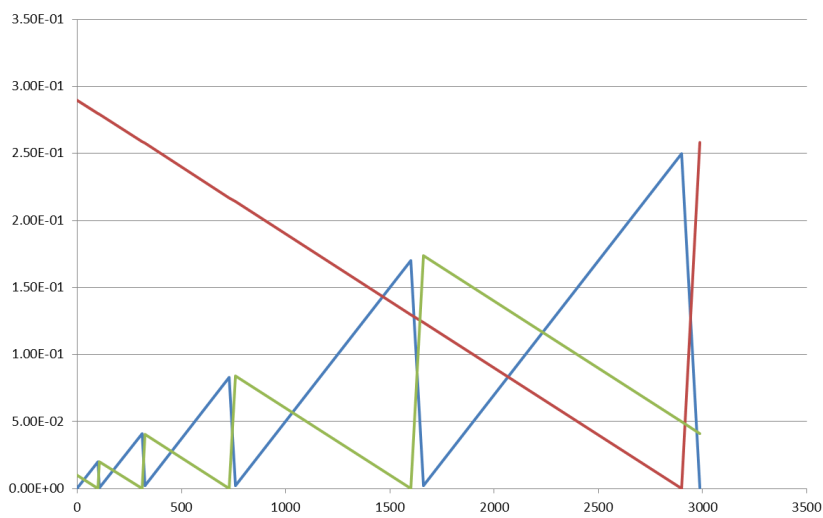


Figure 50. Scénario 3, 3000 itérations, timelapse de 37

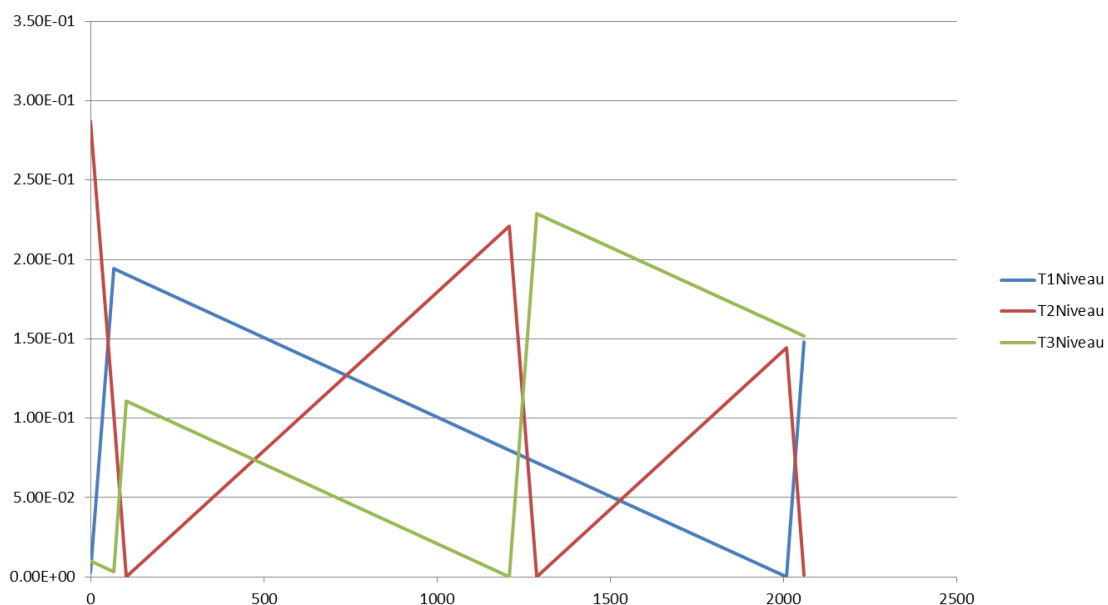


Figure 51. Scénario 4, 2000 itérations, timelapse de 37

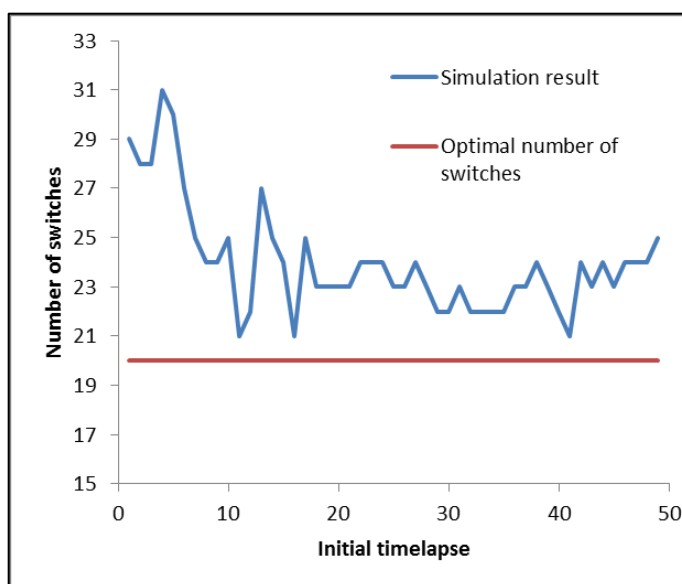


Figure 52. Influence du timelapse initial sur les résultats de simulation du scénario 2 sur 2000 pas

Le premier résultat qui doit être souligné est que la commande holonique a toujours trouvé une solution. Un autre élément important est que la solution optimale a été trouvée dans 6 instances sur 8. Si l'on regarde un peu plus précisément les instances où la solution optimale n'a pas été trouvée, il semble logique de remarquer que ce sont les instances où le nombre optimal de switches est le plus important.

Si l'on considère uniquement les scénaris 1 et 2 (3 et 4 ayant trop peu de switches pour être représentatifs), la performance de la commande holonique est 80 à 90% du temps inférieure à 40% de dégradation de performance. Ceci montre un comportement relativement intéressant de la commande prédictive réactive, et peut laisser entrevoir de

bonnes perspectives en termes de performance en améliorant les algorithmes présentés précédemment.

Table 4. Comparaison des résultats de simulation et de calculs

Scénario	Horizon de calcul (en nombre de pas)	Nombre optimal de switches calculés (MILP)	Meilleurs timelapses	Pourcentage de timelapses de la commande holonique classés en fonction de la dégradation du nombre de switches obtenus					Temps de calcul (sec)	
				0%]0%; 20%]]20%; 40%]]40%; 60%]	>60%	Avec borne sup.	Sans borne sup.
1	1000	10	34	2.04	77.55	12.25	8.16	0	0.31	3.76
	2000	20	29/32/ 34/35/ 39/41/ 43/46	0	67.35	30.61	2.04	0	1053	4338
2	1000	10	29/33/ 34	6.12	69.39	14.29	8.16	2.04	0.75	10.47
	2000	20	16/41	0	75.51	18.37	6.12	0	869	4989
3	1000	6	Tous	100	0	0	0	0	0.37	1.48
	2000	8	Tous	100	0	0	0	0	0.48	2.86
4	1000	2	31-49	38.8	0	0	4.08	57.14	0.2	0.3
	2000	4	21-37	34.7	0	4.08	42.86	18.37	0.11	1.7

A propos des timelapses, aucune corrélation n'a pu être observée entre les meilleurs timelapses entre scénaris et même entre différentes instances (1000 ou 2000 pas de temps) du même scénario. De ce fait, une étude plus approfondie doit être menée de façon à déterminer une manière de déterminer le meilleur timelapse à utiliser dans un contexte d'utilisation en ligne. Une piste serait probablement d'utiliser des notions de simulations en ligne (Cardin and Castagna, 2009), où des simulations en mode « look-ahead » permettraient de tester les différents timelapses sur un horizon représentatif afin de choisir le meilleur.

D'un autre côté, l'algorithme MILP a également toujours trouvé une solution. Toutefois, les temps de calcul sont extrêmement longs (surtout en comparaison des simulations), et augmentent extrêmement rapidement entre 1000 et 2000 pas. Ceci empêche l'utilisation

d'un tel algorithme en ligne, par exemple dans le cas d'une reconfiguration du système. Ce modèle atteint sa limite pour un horizon temporel très petit, à cause du nombre de variables à considérer. Un autre modèle, indexé sur le temps, a été conçu, mais avec des résultats encore pires, ceci étant dû au plus grand nombre de variables à prendre en compte.

Finalement, comme attendu, l'utilisation des résultats pour la borne supérieure a montré de réelles améliorations pour les temps de calcul de l'algorithme, mais dans des proportions où les temps de calcul restent significativement longs.

3.7 Stabilité de la commande

Les sections précédentes ont mis en avant les performances de la solution de commande holonique proposée en termes de flexibilité et d'optimalité. Cette section vise à évaluer la performance de la commande holonique vis-à-vis de la stabilité. En effet, il est important dans cette classe de système de prouver la stabilité de la commande afin de garantir une solution conforme aux exigences du système à tout instant. Cette propriété est particulièrement importante dans le cas du système de commutation d'arrivée, car il s'agit par essence d'un système chaotique.

Chase (Chase et al., 1993) permet l'analyse de la stabilité du système à l'aide de l'opérateur de Markov, appelé opérateur de fonction Perron-Frobenius, défini sur le système autour des points de stabilité appelés points de densité fixe. Ces points sont définis de façon que le système dispose d'orbites périodiques, i.e. un comportement cyclique. Ainsi, une façon de prouver la stabilité du système est de démontrer qu'il est cyclique, c'est-à-dire qu'il existe une séquence d'états et d'événements du système qui se répète à l'infini en l'absence de perturbations. L'approche présentée dans ce document ne permet pas de proposer une démonstration formelle analytique d'un comportement cyclique du système. De ce fait, nous avons opté pour une approche statistique.

3.7.1 Définition des expérimentations

Le modèle Java présenté précédemment est utilisé de nouveau dans une configuration à 3 réservoirs. Dans cette expérimentation, l'état du système a été enregistré à chaque fois qu'un changement de position du serveur était déclenché par le système. Si, à une date donnée, le système rencontre une configuration déjà rencontrée précédemment, alors le comportement du système de contrôle hybride holonique est considéré cyclique. Pour tester la stabilité et la flexibilité du contrôle sur une large gamme de scénaris, une analyse stochastique est réalisée. A chaque répliation, les conditions initiales de niveaux des réservoirs et de débits de fuite sont tirées aléatoirement, suivant les conditions suivantes :

- $\text{Rand}(a;b)$ est défini comme la fonction qui à chaque tirage renvoie un nombre aléatoire compris dans l'intervalle]a;b[, la loi de répartition étant uniforme ;
- $x_1(0) = 0$
- $x_2(0) = \frac{\lfloor \text{Rand}(0;1) * 1000 \rfloor}{1000}$
- $x_3(0) = 1 - x_2(0)$

- $\rho_1 = \frac{\lfloor \text{Rand}(0;1) * 1000 \rfloor}{1000}$
- $\rho_2 = \frac{\lfloor \text{Rand}(0;1) * 1000 \rfloor}{1000} * (1 - \rho_1)$
- $\rho_3 = 1 - \rho_1 - \rho_2$

La simulation renvoie la date où un cycle a été trouvé. Chaque simulation est répliquée 1000 fois de manière à obtenir un échantillon significatif à analyser.

L'horizon de simulation a été étudié également. En effet, toutes les simulations ont un horizon défini a priori, servant comme condition d'arrêt dans le cas où aucun cycle n'était trouvé. Toutefois, certains cycles sont extrêmement longs, et l'étude sur un seul horizon ne permet pas de conclure quant à la performance du contrôle. Il est donc intéressant de faire augmenter l'horizon et d'étudier la répartition des pourcentages de cycles trouvés en fonction du nombre d'itérations considérées (sachant que l'horizon de simulation est défini comme le produit du pas de temps par le nombre d'itérations de simulation).

3.7.2 Résultats et discussion

En regardant les résultats de simulations d'un point de vue général, un point important à noter est qu'aucune violation de contraintes n'a jamais été détectée pour le contrôle holonique, qui a toujours trouvé une solution. Plus en détails, la Figure 53 présente les résultats obtenus par simulation. Tout d'abord, on peut noter qu'il n'a pas été possible d'obtenir des cycles dans 100% des cas avec un temps de calcul raisonnable (fixé à 10h). Cependant, la tendance montre que le contrôle holonique s'en rapproche sensiblement, puisque les échantillons montrent plus de 85% de comportements cycliques détectés pour un horizon de 10^9 itérations. L'évolution de l'horizon montre une étape intéressante à 80% de cycles aux alentours de 10^7 itérations. Ce résultat et la longueur moyenne des cycles d'environ 370.10^3 itérations peuvent être considérés comme des approximations intéressantes de la population totale. La comparaison avec le contrôle classique de (Chase et al., 1993) montre une stabilisation du nombre de scénaris pour lesquels il échoue aux alentours de 45%. Avant que cette valeur ne soit stabilisée, i.e. avant 3.10^6 itérations, les longueurs de cycles peuvent être comparées avec le contrôle holonique; après ce cap, les populations sur lesquelles la moyenne est calculée (i.e. pour (Chase et al., 1993) les instances pour lesquelles il n'a pas échoué) sont trop différentes pour prendre le risque d'une comparaison. Les longueurs de cycles dans la première partie sont finalement très similaires, avec une différence inférieure à 10% en moyenne en faveur du contrôle holonique.

Tous ces résultats montrent que le contrôle holonique présenté dans ce document améliore globalement les performances en comparaison de l'approche classique de (Chase et al., 1993) en terme de flexibilité (toutes les conditions valides permettent au contrôle de trouver une solution respectant les contraintes), engendre une bonne stabilité pour une utilisation sur un terme long et suggère que, avec un algorithme de prise de décision plus optimisé, des solutions performantes pourraient être trouvées et appliquées en ligne.

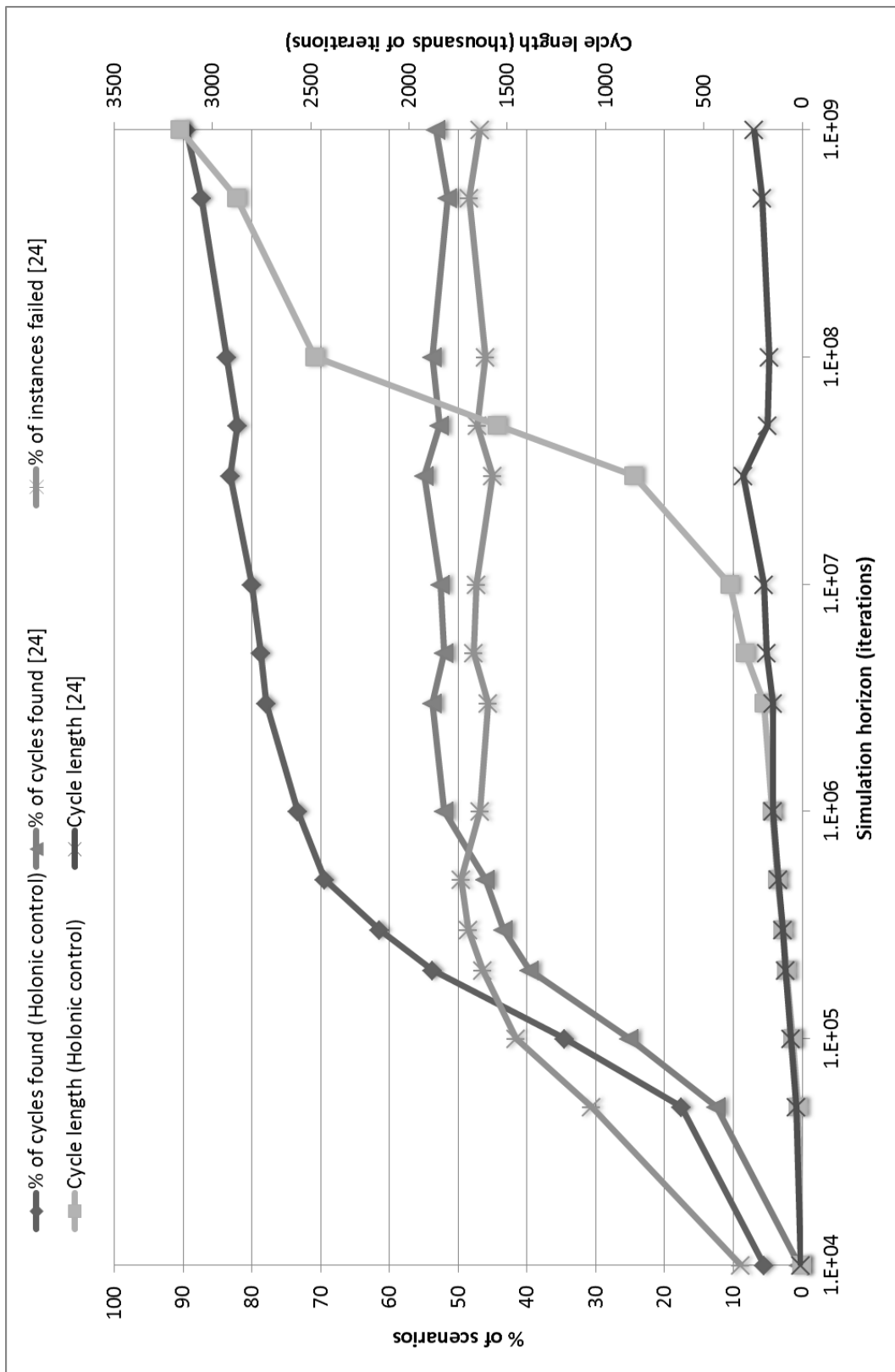


Figure 53. Analyse de stabilité du contrôle holonique.

3.8 Revue du chapitre

Ce chapitre étudie un premier exemple de la mise en œuvre du modèle hybride de contrôle holonique à l'aide d'un exemple classique de la littérature appelé système de commutation d'arrivée. Nous présentons d'abord le modèle utilisé dans la simulation du système, puis nous montrons la mise en œuvre du modèle proposé. Enfin, les résultats sont basés sur l'optimisation, la flexibilité et la stabilité. L'optimisation du système est basée sur le nombre de commutations et comparé avec le modèle d'ordonnement optimal du nombre de changements de position du serveur. Pour la flexibilité, nous montrons comment l'approche holonique a permis très facilement de passer d'un modèle à trois réservoirs à un modèle à dix réservoirs. Enfin, nous montrons une remarquable stabilité du pilotage holonique par rapport à l'approche classique de (Chase et al., 1993).

Chapitre 4

4 Cas d'étude de centrale thermique à cycle combiné

Un cas d'étude a été choisi autour d'un process industriel de génération d'électricité, en particulier un Process à Cycle Combiné (Combiné Cycle Plant – CCP) (De Mello et al., 1994; Rowen, 1992), qui est un système hybride où le comportement des turbines représente la dynamique continue et le séquençement des turbines la dynamique discrète. Un modèle de contrôle holonique et une émulation du CCP physique ont été développés en Java et un benchmark de la littérature a été choisi de manière à valider l'efficacité de l'approche.

Ce chapitre commence avec la description du CCP, suivie par la description des modèles utilisés pour la représentation des turbines dans l'émulation du CCP. Dans la troisième partie, le modèle de contrôle holonique est décrit, suivi par la définition des expérimentations. Finalement, des résultats de comparaison avec la littérature sont présentés.

4.1 Description du process de cycle combiné

Un CCP est un process thermodynamique ayant pour objectif de produire de l'énergie électrique grâce à deux différents cycles (Zapata Madrigal, 2011). Le premier est un cycle ouvert alimenté en air et essence. La turbine à gaz est un moteur à combustion interne, avec une chambre de combustion alimentée en essence (diesel ou essence) et air, Figure 54. Cette combustion génère du gaz en capacité suffisante pour entraîner une turbine couplée à un générateur synchrone. Durant ce cycle, presque 65% de l'énergie est perdue, principalement dissipée en chaleur, ce qui induit un coût très élevé de l'énergie (en €/kWh).

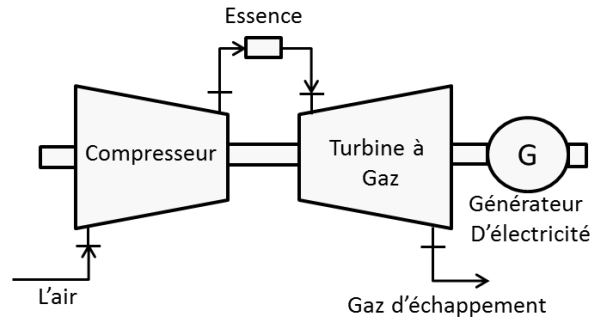


Figure 54. La turbine à Gaz.

Un second cycle peut donc être ajouté. C'est un cycle fermé de vapeur qui récupère la chaleur du gaz produit lors du premier cycle et génère de la vapeur surchauffée en utilisant une chaudière de récupération de vapeur (échangeur de chaleur qui permet de récupérer l'énergie thermique d'un écoulement de gaz chaud - Heat Recovery Steam Generator - HRSG), vapeur qui est ensuite envoyée vers une turbine à vapeur. La turbine à vapeur est un moteur à combustion externe, qui fonctionne en utilisant la vapeur générée par le HRSG. La vapeur est générée avec suffisamment de puissance pour entraîner en rotation une turbine à vapeur couplée à un générateur. La sortie de vapeur de la turbine à vapeur passe ensuite au travers d'un condenseur et est recyclée, établissant de fait un cycle d'eau fermé, comme le montre la Figure 55. Ce second cycle aide à améliorer l'efficacité du process global et diminue le coût du kWh.

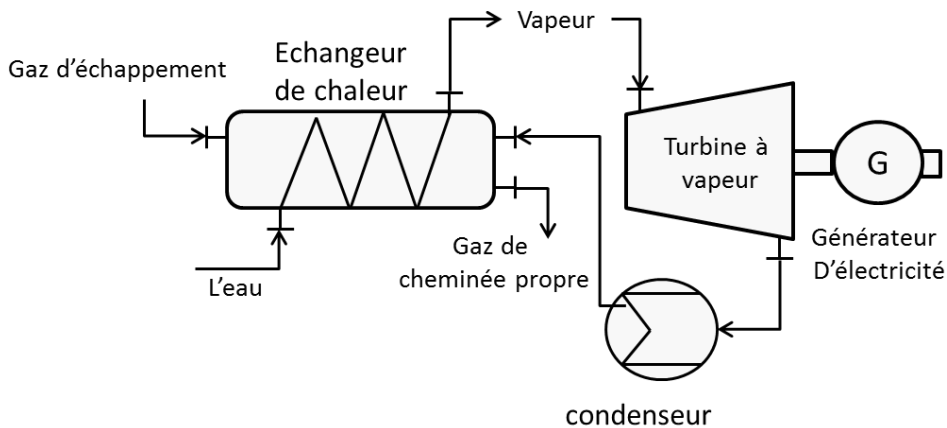


Figure 55. La turbine à vapeur.

L'arrangement du cycle combiné donne 50% d'énergie en plus et un rendement de conversion de 60%. La configuration basique d'un CCP est présentée Figure 56.

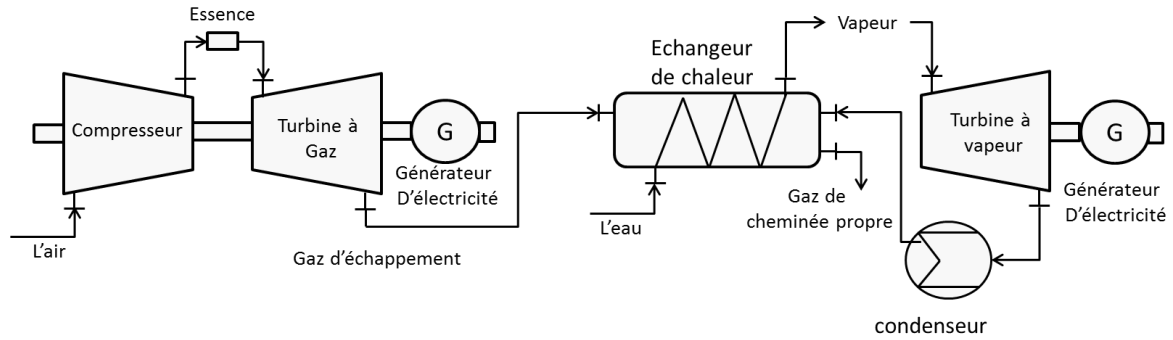


Figure 56. Configuration basique d'un CCP.

4.2 Modèles du cas d'étude

L'objectif de cette section est de décrire les modèles utilisés dans l'émulation du CCP. Les modèles proviennent de la littérature: le modèle de turbine à gaz est présenté en détail dans (Rowen, 1983) et dénommé Heavy Duty Gas Turbine (HDGT) dans la suite, le modèle de turbine à vapeur est lui présenté en détail dans (De Mello et al., 1994).

4.2.1 Modèle de turbine à gaz

Les HDGT sont des turbines à gaz spécifiquement conçues pour une génération de haute puissance, ayant pour caractéristique une haute durée de vie et un taux de disponibilité supérieur aux autres turbines à gaz (Rowen, 1983). Les HDGT sont composés de 3 éléments principaux: des compresseurs centrifuges axiaux multi-étages, des chambres de combustion annulaires et des turbines à écoulement axial. De l'air à conditions atmosphériques est injecté dans le compresseur après passage dans les filtres à air de l'entrée. Le compresseur multi-étages augmente la vitesse, la pression et la température de l'air avant qu'il n'atteigne la chambre de combustion et les zones haute pression de la turbine.

Chaque étage de compression comprend un rang de pales de rotor et d'aubes de stator. L'air comprimé à haute température et haute pression poursuit sa voie jusqu'à la chambre de combustion. Cette chambre est par nature un chauffage dans lequel l'essence est brûlée pour augmenter la température en restant à une pression globalement constante. A peu près un tiers de l'air extrait du compresseur est mélangé avec l'essence pour la combustion. Le volume d'air restant est mélangé avec le produit de combustion pour devenir le flux d'entrée de la turbine, qui est de ce fait déjà à la température d'admission de la turbine. Le flux est ensuite détendu dans les différents étages de la turbine, ce qui entraîne le compresseur et le générateur. Enfin, le flux est guidé à travers la tubulure d'échappement vers un second environnement, qui sera soit l'air ambiant ou le HRSG dans le cas d'un cycle combiné.

Les turbines à gaz suivent le cycle de Brayton. La Figure 57 présente un cycle de Brayton standard en coordonnées température-entropie. L'air est introduit au point 1 dans des conditions atmosphériques et est comprimé par le compresseur jusqu'au point 2. La combustion va alors faire augmenter la température jusqu'au point 3, où le produit de la

combustion et l'air de sortie du compresseur entrent dans la turbine et se détendent jusqu'au point 4. Dans cette figure, la perte de pression dans les filtres à air et la chambre de combustion sont négligées, i.e. les phases 2-3 et 4-1 sont supposées isobares. Les phases à l'intérieur du compresseur et de la turbine sont irréversibles et non-isentropiques. Toutefois, la Figure 57 présente également les phases isentropiques supposées.

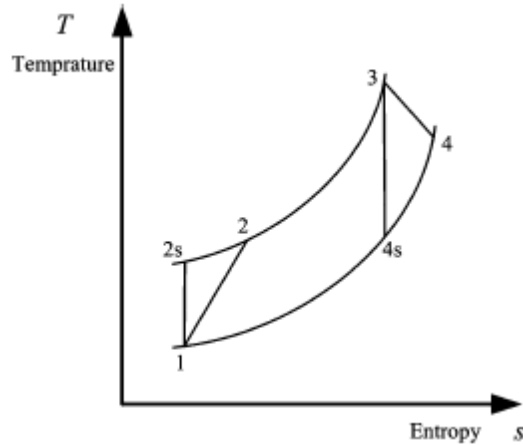


Figure 57. Le cycle de Brayton.

La Figure 58 présente le modèle HDGT de (Rowen, 1983). Le comportement de la turbine est reflété par deux quantités: le couple de sortie T_{mech} et la température des gaz d'échappement $T_{exhaust}$.

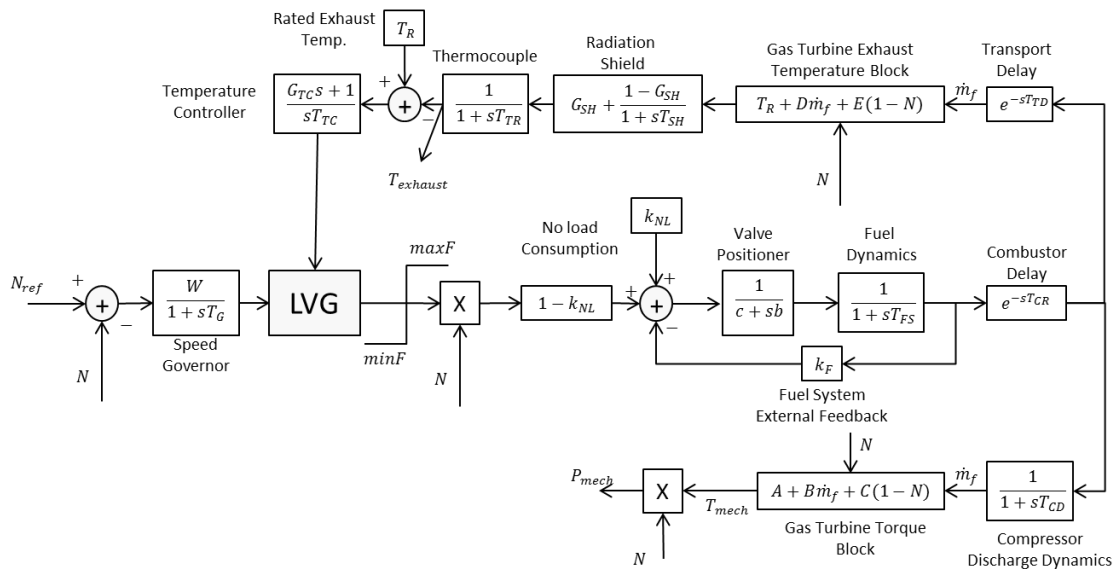


Figure 58. Modèle HGDT de (Rowen, 1983)

Après calculs, les équations peuvent être exprimées par le schéma bloc présenté Figure 58, d'où l'on déduit que:

$$P_{Gpu} = A + B\dot{m}_{fpu} \quad \text{à vitesse nominale (1)}$$

$$A = \frac{\dot{m}_n T_1}{P_{Gn}} \left\{ C_{ph} n_t \left(1 - \frac{1}{x_h} \right) - \frac{x_c - 1}{n_c} \times \left[C_{pc} - C_{ph} n_t \left(1 - \frac{1}{x_h} \right) \right] \right\} \quad (2)$$

$$B = \frac{n_{comb} n_t H \dot{m}_{fn}}{P_{Gn}} \left(1 - \frac{1}{x_h} \right) \quad (3)$$

où A et B sont les coefficients du couple de sortie de la Figure 58, et \dot{m}_n et \dot{m}_{fn} les débit nominaux d'air et d'essence. P_{Gpu} est la puissance de sortie par unité, qui est proportionnelle au couple par unité. La puissance nominale de la turbine est la base de calcul de l'unité. A vitesse nominale, la température d'échappement peut être calculée par l'équation:

$$T_4 = T_R - D(1 - \dot{m}_{fpu}) \quad \text{à vitesse nominale} \quad (4)$$

$$D = n_{comb} \frac{H}{C_{ph}} \frac{\dot{m}_{fn}}{\dot{m}_n} \left[1 - \left(1 - \frac{1}{x_h} \right) n_t \right] \quad (5)$$

où D est le coefficient du bloc de température d'échappement de la Figure 58, T_R est la température d'échappement nominale du HDGT et T_E est la température d'échappement de la turbine à gaz.

Pour identifier les paramètres de la turbine, une condition opérationnelle typique est choisie pour dériver les rendements de la turbine et du compresseur, puis tous les paramètres cités ci-dessus sont calculés grâce aux données disponibles issues des différents essais.

Pour les moteurs terrestres, les données de performance sont fréquemment citées dans les conditions de l'ISO d'un seul point (i.e., 1 atm pression ambiante, l'humidité relative 15 à 60%). Ces paramètres (Tavakoli et al., 2009) sont fournis pour une turbine de 100 MW dans la Table 5. La turbine est une HDGT à arbre unique alimentée au gaz naturel.

En utilisant le modèle de la Figure 58 et les paramètres de la Table 5, un modèle HDGT a été programmé en utilisant Matlab-Simulink, Figure 59. Ce modèle n'a pas directement été utilisé pour l'émulation, mais plutôt utilisé pour la validation du modèle d'émulation discrétisé. En effet, les schémas-blocs Simulink et quelques réponses caractéristiques du système (à un échelon principalement) peuvent être trouvés dans la littérature. L'idée est donc de comparer les résultats obtenus avec les modèles d'émulation discrétisés, programmés en Java, à ceux modélisés sous Simulink de manière à valider la discrétisation.

Pour construire ce modèle Java, les équations (1) et (2) ont été discrétisées en temps explicitement, en suivant la structure de contrôle de la Figure 58 et les paramètres de la Table 5. Pour la validation du modèle Java, un échelon a été imposé pour amener la turbine de 0% à 50% et de 0% à 100% de sa puissance maximale, Figure 60. Seulement de faibles différences ont pu être détectées entre les modèles, et ce seulement durant les phases transitoires. Ces phases étant relativement courtes (de l'ordre de la minute) comparées à la dynamique du système (de l'ordre de l'heure), et ces phases n'étant pas

prises en compte dans les résultats traités dans la suite de ce document, ces différences ont été considérées comme négligeables.

Table 5. Paramètres du modèle HGDT d'une turbine 100 MW.

Paramètre	Symbole	Valeur
Gain du régulateur de vitesse	W	25
Constante de temps du régulateur de vitesse t	T_G	0.05
Limite maximale de la demande en carburant	maxF	1.5
Limite minimale de la demande en carburant	minF	-0.13
Consommation de carburant à vide	K_{NL}	0.24
Constante de temps du positionneur de vanne (s)	B	0.04
Constante de temps du système d'alimentation en carburant (s)	T_{fs}	0.26
Gain de la boucle d'asservissement du système d'alimentation en carburant	K_F	0
Retard du système de combustion (s)	T_{CR}	0.005
Retard du système de transport de la turbine et de l'échappement (s)	T_{TD}	0.04
Constante de temps du retard du refoulement du compresseur (s)	T_{CD}	0.16
Paramètre du bloc de couple de la turbine à gaz	A	-0.158
Paramètre du bloc de couple de la turbine à gaz	B	1.158
Paramètre du bloc de couple de la turbine à gaz	C	0.5
Paramètre du bloc de température d'échappement de la turbine à gaz (°C)	D	413
Paramètre du bloc de température d'échappement de la turbine à gaz (°C)	E	313
Paramètre de l'écran anti-rayonnements	G_{SH}	0.85
Constante de temps de l'écran anti-rayonnements (s)	T_{SH}	12.2
Constante de temps du thermocouple (s)	T_{TR}	1.7
Température d'échappement (°C)	T_R	522
Paramètre du contrôleur de température	G_{TC}	3.3
Constante d'intégration du contrôleur de température (°C)	T_{TC}	250

4.2.2 Modèle de turbine à vapeur

Le HRSG (chaudière de récupération de vapeur) réagit aux changements du débit W , et de la température T_E des échappements de la turbine à gaz. La puissance mécanique développée par la turbine à vapeur suit les changements de l'énergie des échappements de la turbine à gaz avec retard, que ce soit à haute ou basse pression.

Il est vraisemblable que la contribution à la puissance mécanique des deux chaudières (haute et basse pression) puisse être approximée par un modèle à deux constantes de temps. Le gain entre l'énergie de l'échappement de la turbine à gaz et la sortie de vapeur de la turbine à vapeur est en général une fonction non-linéaire qui peut être dérivée de mesures réalisées en régime permanent à travers la plage de charge. Le modèle HRSG choisi et le modèle de turbine à vapeur ont été combinés dans (De Mello et al., 1994) en un seul modèle simplifié, représenté par le schéma bloc de la Figure 61. L'utilisation de ce modèle simplifié empêche de découpler le HSRG et la turbine à vapeur, ce qui reste une approximation pour la conception du cas d'étude, mais la simplification apportée par ce modèle évitera d'amener d'autres approximations avec un modèle plus complexe, ce qui sera finalement bénéfique pour le résultat final.

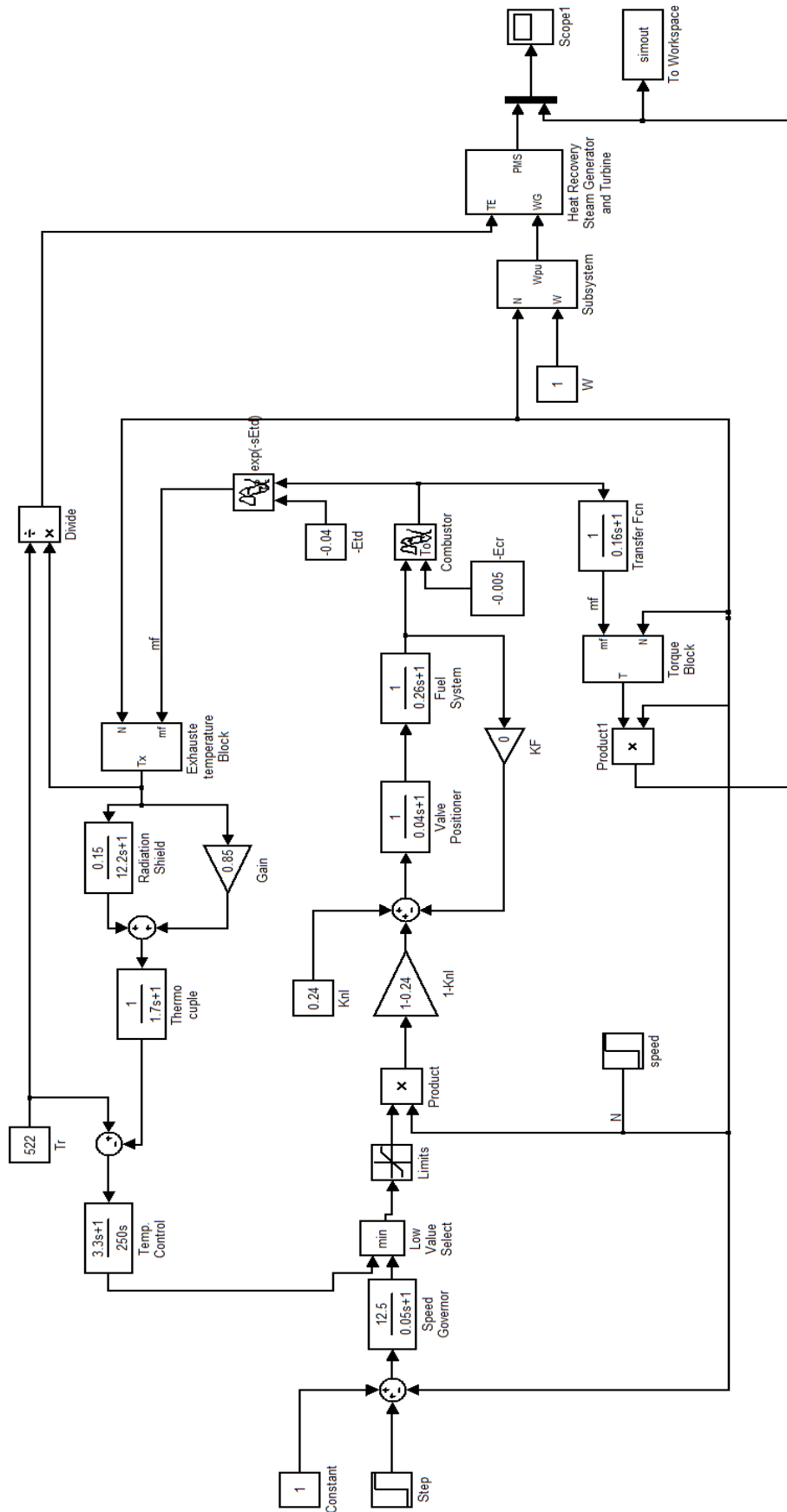


Figure 59. Modèle HDGT issu de Matlab – Modèle de référence.

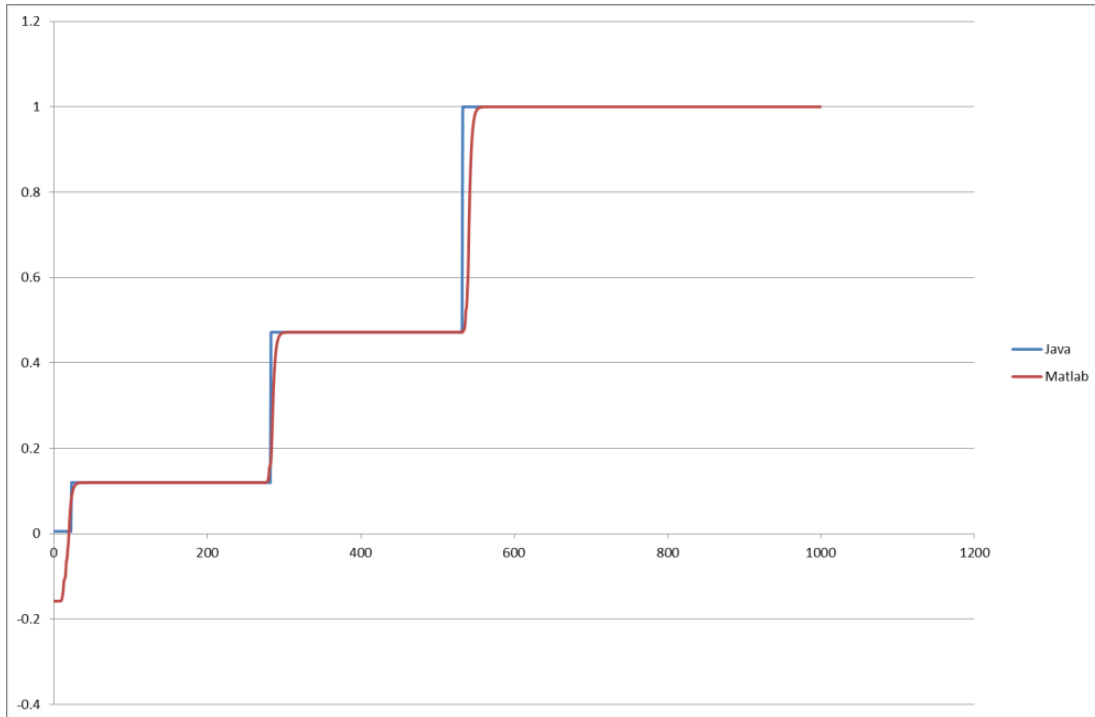


Figure 60. Comparaison de la simulation du modèle Java et du modèle Matlab de la turbine.

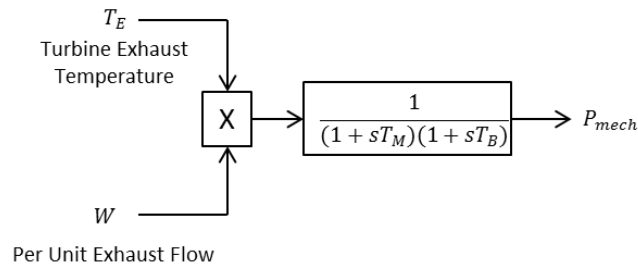


Figure 61. Modèle de Turbine à vapeur et d'HRSG simplifié

4.2.3 Modélisation Java des turbines

Le CCP a été émulé en utilisant le langage Java. Huit classes ont été créées en plus de la classe principale: GasTurbine, VapeurTurbine, GasTurbineSim, VapeurTurbineSim, Resource, Product, Order et service (Figure 62).

Les modèles HDGT et de turbines à vapeur ont été programmés dans les classes GasTurbine et VapeurTurbine, et une méthode "Simulation" a été créée qui contient toutes les équations de turbines à gaz et à vapeur, ce qui fait que ces classes représentent la partie physique des turbines. Les classes GasTurbineSim et VapeurTurbineSim sont une abstraction des classes GasTurbine et VapeurTurbine et représentent la partie logique des turbine à gaz et à vapeur. A l'intérieur de ces classes, la méthode "descriptionmodel" contient un graphe représentant l'ensemble des états possibles des turbines.

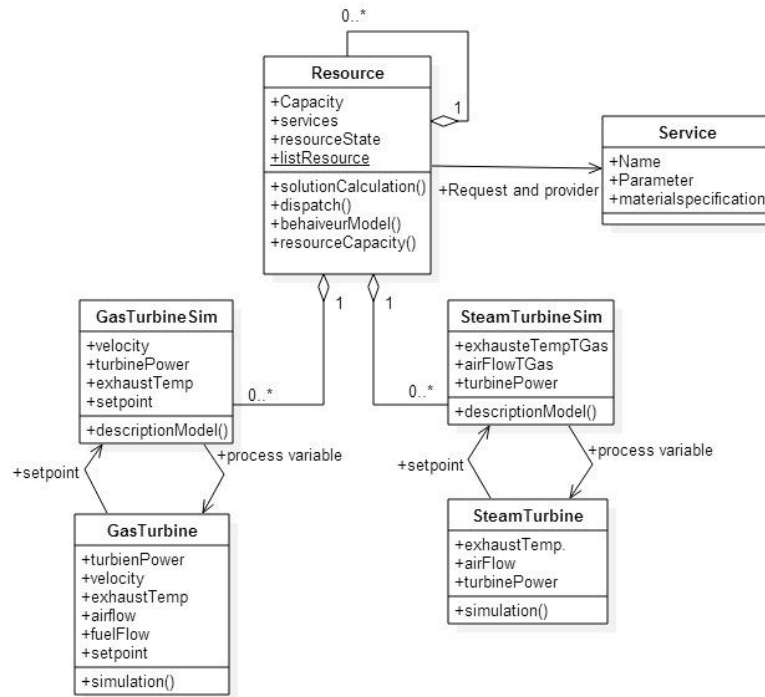


Figure 62. Diagramme de classe du programme Java de CCP.

La classe Resource est la classe où toutes les capacités des ressources (services, capacités, coût, etc.) sont définies. Elle contient aussi les méthodes "dispatch" et "solutionCalculation" qui font partie du mécanisme de négociation pour déterminer la meilleure configuration de production.

Enfin, la classe "Service" définit la structure des services fournis et demandés. Dans cette application, le service défini dans l'émulation du CCP est la génération d'électricité avec des spécifications de paramètres telles que la fréquence, le type de carburant utilisé, etc.

4.3 Application de H²CM sur un Process à cycle combiné

La première étape pour implanter H²CM sur un CCP est de définir tous les holons du système. Cette définition est faite suivant la structure fractale proposée dans le Chapitre 2, où les turbines à gaz et à vapeur sont définies comme holons atomiques. Un arrangement 4x2 (4 turbines à gaz pour 2 turbines à vapeur) a été choisi. Cet arrangement et la connexion des unités dans une configuration 2 x 1 forment un holon composé. Donc, le système contient deux holons composés de même structure qui seront dénommés holons ressources 2 et 3, qui à leur tour forment un troisième holon dénommé holon ressource 1, comme présenté dans la Figure 63. Le holon produit est défini par le service de génération électrique de chaque turbine, service qui peut être paramétré (fréquence, carburant, etc.). Il y a une distinction entre les produits des turbines à gaz et des turbines à vapeurs : les turbines à gaz génèrent de l'énergie et de la chaleur alors que les turbines à vapeur ne génèrent que de l'énergie. De plus, les turbines à vapeur nécessitent la chaleur générée par les turbines à gaz pour fonctionner.

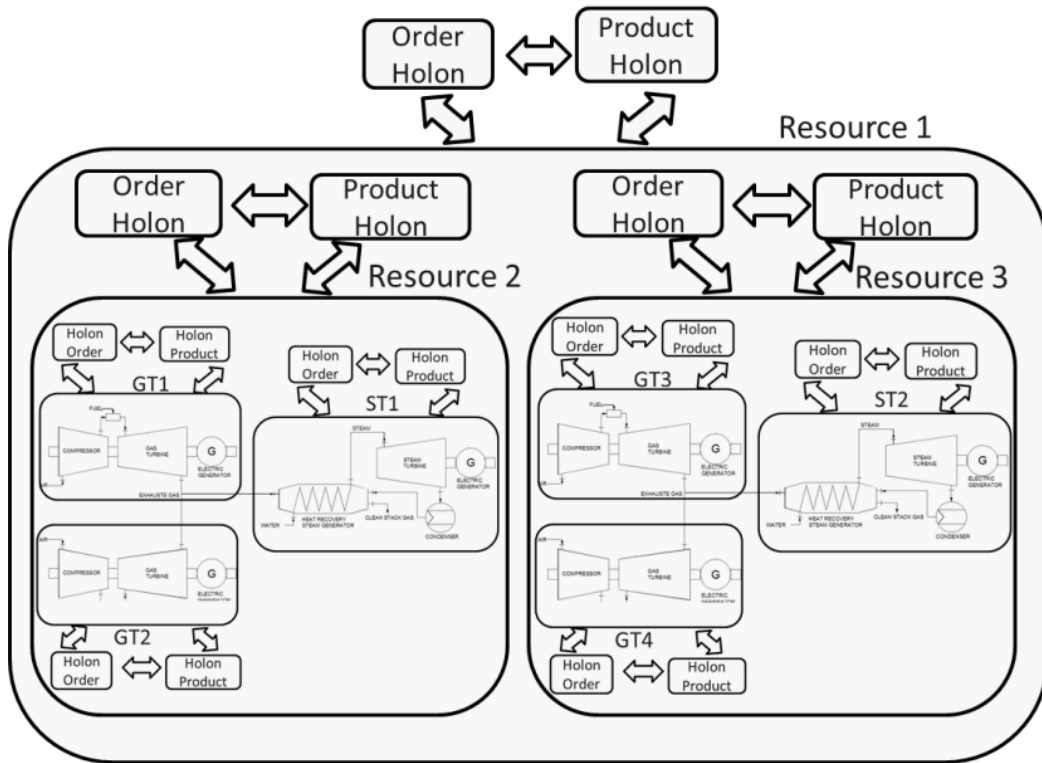


Figure 63. Modélisation holonique du CCP 4x2.

Au cours de la planification, les holons négocient entre eux comme présenté dans le diagramme de séquence de la Figure 64. Chaque ressource composée crée un nouveau sous-ordre paramétré avec ses besoins (séquence n°7), les holons atomiques, selon leur capacité et disponibilité, calculent une solution (séquence n°12) et envoient une proposition (séquence n°14). Le holon composé reçoit toutes les propositions et évalue la meilleure solution (séquence n°16), et génère donc le planning et l'holararchie qui va exécuter le planning.

L'évaluation de la solution du holon composé (séquence n°16) est réalisée en suivant l'algorithme de la Figure 65. La nomenclature utilisée dans cet algorithme est le suivant:

- t : temps de la planification (en heures);
- $D_R(t)$: Demande de la ressource au temps t ;
- R_i : Ressource i à l'intérieur de la ressource composée R ;
- $C_{R_i}(t)$: Capacité de la ressource i ;
- N : Nombre de ressources disponible dans R ;
- $C_R = \sum_1^N C_{R_i}$;
- $L_{R_i}(t)$: Charge de la ressource i au temps t ;
- G_i : Turbine à gaz i ;
- $L_{G_i}(t)$: Charge de la turbine à gaz i ;
- S_j : Turbine à vapeur j ;
- $L_{S_j}(t)$: Charge de la turbine à vapeur j ;
- $\$R$: Coût de la ressource R ;

- t_1 : Date initiale de planification;
- t_2 : Date finale de planification.

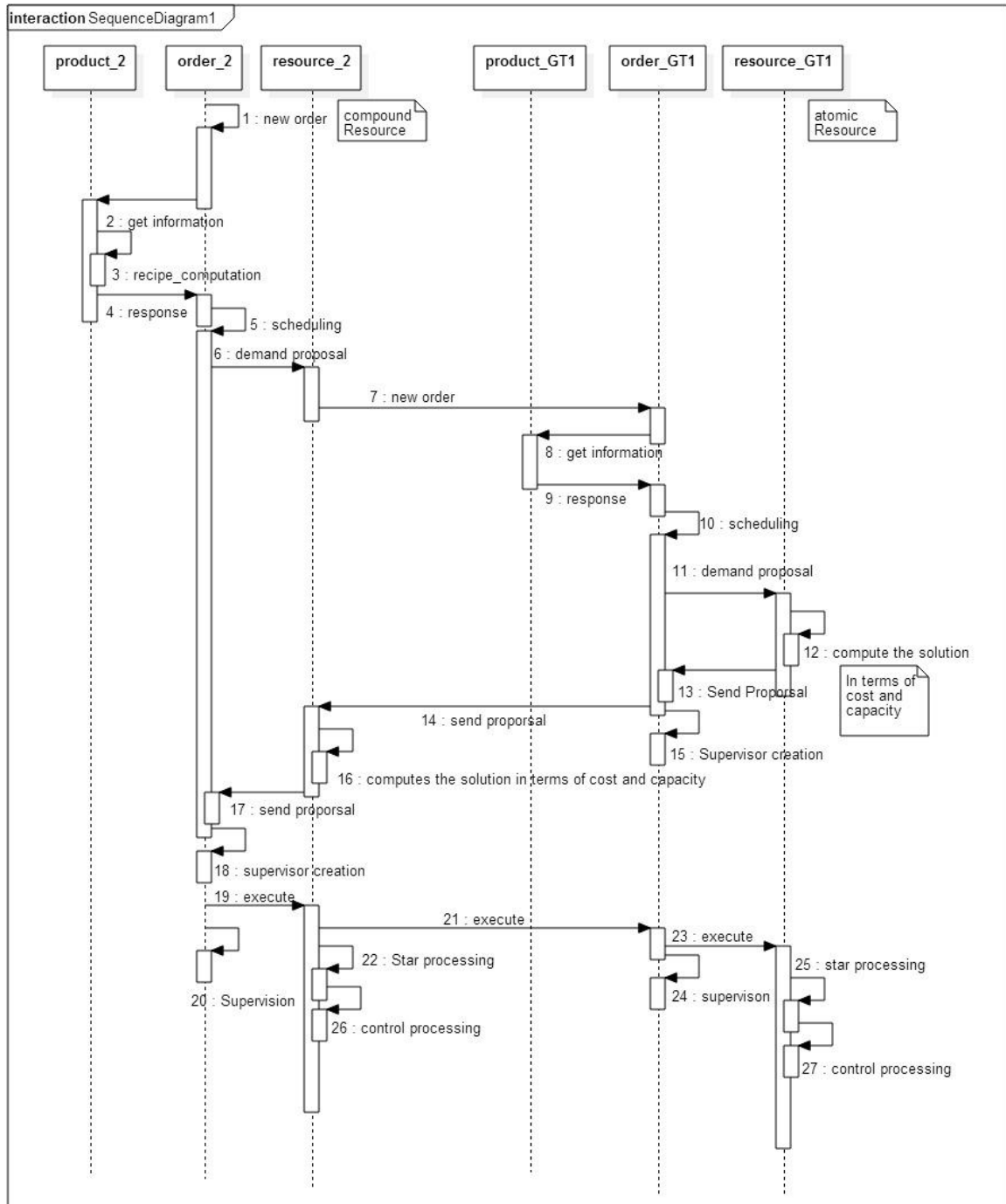


Figure 64. Diagramme de séquence de planification du CCP.

Chaque fois que l'algorithme est exécuté, un paramétrage est nécessaire avec la demande en énergie adéquate, la période à planifier et les ressources concernées. Il est globalement divisé en deux cas. Dans le premier cas, le holon composé est lui-même composé de holons composés. La planification des ressources est alors une fonction de trois variables: capacité, disponibilité et coût des ressources. Tout d'abord, la ressource la moins

coûteuse est choisie et la capacité disponible est directement allouée. Le second cas apparaît quand la ressource est composée de ressources atomiques. Dans ce cas, la charge est distribuée uniformément entre les ressources disponibles, en considérant toutefois leurs capacités minimales et maximales. L'objectif d'utiliser toutes les ressources atomiques du holon composé est d'augmenter la réactivité du système: en cas de défaillance, les ressources atomiques sont directement disponibles et démarrées, donc il n'y a plus qu'à ajuster leur consigne à temps pour compenser la défaillance. Dans l'autre cas, avoir besoin de démarrer une turbine complètement arrêtée ne permettrait pas de réagir suffisamment rapidement à la défaillance, le temps de démarrage se comptant en dizaines de minutes alors que la réponse à un échelon d'une turbine déjà démarrée se fait en quelques secondes.

Enfin, le superviseur utilisé dans l'implantation de H²CM a pour fonction de vérifier le statut de chaque ressource. Au moment où une ressource change d'état (par exemple quand une ressource change de opérationnelle à dégradée), le superviseur détecte le nouveau statut (et donc dans ce cas une défaillance) et exécute une replanification de la production paramétrée par la période et les ressources à replanifier.

4.4 Définition des expérimentations

De manière à vérifier expérimentalement les résultats et impacts du modèle proposé, une expérimentation est proposée, avec une usine produisant de l'énergie sur un horizon de 24h et deux défaillances lors de cette production. Il est important ici de clarifier que le propos ici n'est pas d'évaluer l'optimalité des algorithmes proposés ci-dessus, mais bien d'évaluer la conformité d'H²CM avec les attributs du modèle holonique tels que présentés précédemment, i.e. bonnes performances sur un HCS, réaction adéquate aux défaillances et flexibilité du système de contrôle. Les caractéristiques de capacités et coûts de chaque turbine utilisée dans le cas d'étude sont présentées dans la Table 6. Ces données, ainsi que le scénario, est issu des travaux de (Zapata Madrigal, 2011). Cela nous permettra d'avoir un point de comparaison raisonnable sur la performance du système par rapport à une approche classique et ainsi voir si le comportement des algorithmes de décision est cohérent.

De manière à tester l'hypothèse de fonctionnement correct de H²CM, une méthode est définie pour vérifier la planification, la résolution d'une défaillance due à la modification de la capacité d'une ressource et la résolution défaillance due à l'arrêt d'une ressource. De ce fait, le plan expérimental suivant a été établi:

- a. Définition d'une mission sur une période de 24 heures (24 périodes d'une heure);
- b. Planification de la production et formation des holarchies;
- c. Génération des défaillances, Table 7;
- d. Re-planification de la production.

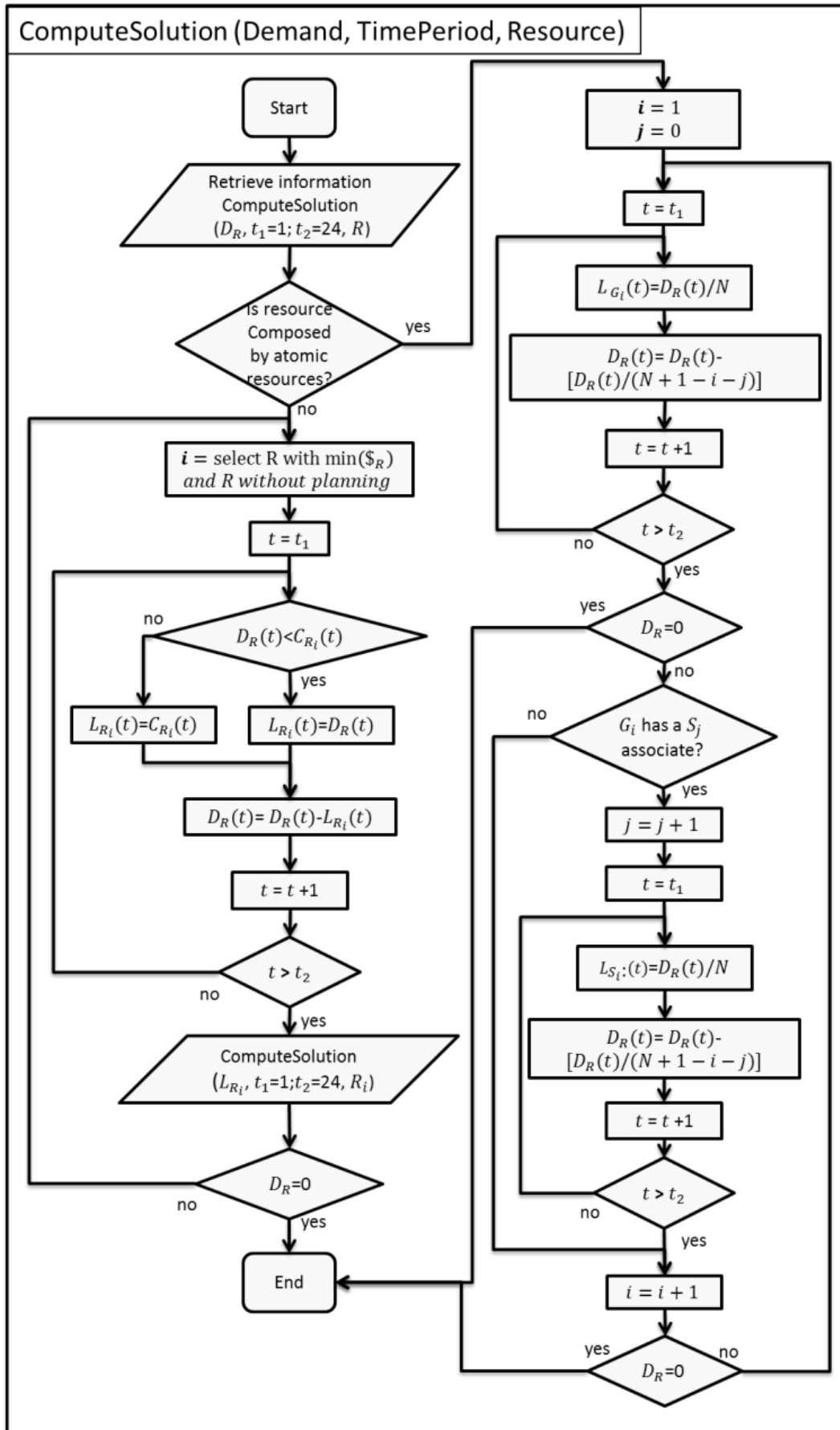


Figure 65. Algorithm « Compute best solution ».

Table 6. Capacités des ressources.

Holon	Coût (US\$/MW)	Maximum Capacité (MW)	Minimum Capacité (MW)
Turbine à gaz 1 (GT1)	1000	100	15
Turbine à gaz 2 (GT2)	1100	100	15
Turbine à gaz 3 (GT3)	1200	100	15
Turbine à gaz 4 (GT4)	1300	100	15
Turbine à vapeur 1 (ST1)	400	100	10
Turbine à vapeur 2 (ST2)	500	100	10

Table 7 Défaillances de turbines.

Défaillance	Période	Ressource concernée	Type de défaillance	Description
1	11	ST1	Dégradation	De 100% à 80% de capacité
2	19	GT2	Arrêt	de 100% à 0% de capacité

4.5 Résultats

4.5.1 Planning

Le premier résultat concerne la planification de la demande et la création d'holarchies. Sur la Table 8, la demande totale d'énergie du système peut être lue au niveau de r1 (qui représente l'ensemble de l'usine). L'assignation de la charge à chaque ressource par période et la manière dont sont formées les holarchies peuvent également être lues dans les colonnes suivantes. Par exemple, durant les périodes 5 à 11, seules les ressources r1, r2, GT1, GT2 et ST1 forment l'holarchie, alors que durant les périodes 12 à 21, l'holarchie consiste en l'ensemble des ressources. Ce planning résulte de l'instanciation de l'algorithme présenté Figure 65. En effet, r2 (ressource composée) a un coût de production plus bas que r3, donc une plus grande charge est assignée à r2. L'allocation de la charge au sein de r2 est faite de manière uniforme entre les ressources atomiques.

Table 8. Planning de production.

Période	r1	r2	r3	GT1	GT2	ST1	GT3	GT4	ST2
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5.00	90.00	90.00	0.00	30.00	30.00	30.00	0.00	0.00	0.00
6.00	90.00	90.00	0.00	30.00	30.00	30.00	0.00	0.00	0.00
7.00	100.00	100.00	0.00	33.33	33.33	33.33	0.00	0.00	0.00
8.00	150.00	150.00	0.00	50.00	50.00	50.00	0.00	0.00	0.00
9.00	150.00	150.00	0.00	50.00	50.00	50.00	0.00	0.00	0.00
10.00	250.00	250.00	0.00	83.33	83.33	83.33	0.00	0.00	0.00
11.00	250.00	250.00	0.00	83.33	83.33	83.33	0.00	0.00	0.00
12.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
13.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
14.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
15.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
16.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
17.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
18.00	350.00	300.00	50.00	100.00	100.00	100.00	16.67	16.67	16.67
19.00	350.00	300.00	50.00	100.00	100.00	100.00	16.67	16.67	16.67
20.00	350.00	300.00	50.00	100.00	100.00	100.00	16.67	16.67	16.67
21.00	350.00	300.00	50.00	100.00	100.00	100.00	16.67	16.67	16.67
22.00	200.00	200.00	0.00	66.67	66.67	66.67	0.00	0.00	0.00
23.00	200.00	200.00	0.00	66.67	66.67	66.67	0.00	0.00	0.00
24.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

4.5.2 Flexibilité

Des défaillances sont intentionnellement introduites dans le système pour observer son comportement. Les réactions du système peuvent être observées en détail dans la Table 9. La première défaillance est une dégradation de la capacité de ST1 de 100 à 80MW durant les 20 premières minutes de la période 11. La réaction du système est de compenser la perte de production en utilisant directement les autres ressources atomiques du holon composé r2, et donc assigner la charge manquante à GT2. Cette réaction démontre l'autonomie du holon, qui a pu évaluer la possibilité de réagir seul sans en référer à r1, et la réactivité du système de contrôle vis-à-vis de la défaillance. Etant donné que la défaillance ne dure que 20 minutes, la capacité de ST1 est restaurée dès l'heure suivante.

Lors de la seconde défaillance, un arrêt complet de production de GT2 apparaît à la période 19. Le holon r2 n'a pas la capacité disponible pour compenser de manière autonome la perte de production. De ce fait, r2 requiert la collaboration du holon r3 à travers le holon r1. Le système décide alors d'augmenter l'assignation de charge de r3.

La Figure 66 présente de manière graphique la demande, le résultat de la planification et de la réponse aux défaillances, ainsi que les différences entre demande et énergie fournie. De courtes périodes de surproduction sont à noter aux moments de grands changements de la demande. Ces surproductions sont dues au temps de réponse des turbines qui nécessitent de démarrer celles-ci avant l'évolution effective de la demande.

4.5.3 Performance

Un tableau comparatif des résultats entre les coûts induits par la planification holonique et la planification issue de l'approche de (Zapata Madrigal, 2011) est présenté Table 10. Bien qu'aucun de ces travaux n'ait réellement cherché à optimiser le coût induit, la comparaison est intéressante puisque l'approche holonique a des coûts tout à fait comparables à ceux de (Zapata Madrigal, 2011), voire même quelquefois meilleurs, bien que les résultats soient sensiblement différents au niveau du planning obtenu.

Enfin, huit configurations d'usines plus grandes ont été créées, jusqu'à atteindre des tailles d'usine pratiquement irréalistes. La demande a été systématiquement ajustée en conséquence. Les temps de calcul de la planification initiale ont été enregistrés sur un processeur Intel i7- 2.0 GHz et une mémoire RAM de 8Go et sont présentés Table 11. Les résultats montrent que ces temps augmentent évidemment avec la taille de l'usine, mais restent des temps très courts vis-à-vis de la dynamique du système, ce qui tend à montrer que H²CM est une solution utilisable pour le contrôle des HCS.

Table 9. Replanification du système après défaillance.

Période	r1	r2	r3	GT1	GT2	ST1	GT3	GT4	ST2
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5.00	90.00	90.00	0.00	30.00	30.00	30.00	0.00	0.00	0.00
6.00	90.00	90.00	0.00	30.00	30.00	30.00	0.00	0.00	0.00
7.00	100.00	100.00	0.00	33.33	33.33	33.33	0.00	0.00	0.00
8.00	150.00	150.00	0.00	50.00	50.00	50.00	0.00	0.00	0.00
9.00	150.00	150.00	0.00	50.00	50.00	50.00	0.00	0.00	0.00
10.00	250.00	250.00	0.00	83.33	83.33	83.33	0.00	0.00	0.00
11.00	250.00	250.00	0.00	83.33	86.67	80.00	0.00	0.00	0.00
12.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
13.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
14.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
15.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
16.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
17.00	450.00	300.00	150.00	100.00	100.00	100.00	50.00	50.00	50.00
18.00	350.00	300.00	50.00	100.00	100.00	100.00	16.67	16.67	16.67
19.00	350.00	200.00	150.00	100.00	0.00	100.00	50.00	50.00	50.00
20.00	350.00	200.00	150.00	100.00	0.00	100.00	50.00	50.00	50.00
21.00	350.00	200.00	150.00	100.00	0.00	100.00	50.00	50.00	50.00
22.00	200.00	200.00	0.00	100.00	0.00	100.00	0.00	0.00	0.00
23.00	200.00	200.00	0.00	100.00	0.00	100.00	0.00	0.00	0.00
24.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 10. Coûts de production en US\$.

Période	1	2	3	4	5	6	7	8
(Zapata Madrigal, 2011)	0	0	0	0	102500	102500	94165	135840
H ² CM	0	0	0	0	75000	75000	83333	125000
Gain	0%	0%	0%	0%	27%	27%	12%	8%

Période	9	10	11	12	13	14	15	16
(Zapata Madrigal, 2011)	135840	219165	219165	400000	400000	400000	400000	400000
H ² CM	125000	208333	208333	400000	400000	400000	400000	400000
Gain	8%	5%	5%	0%	0%	0%	0%	0%

Période	17	18	19	20	21	22	23	24
(Zapata Madrigal, 2011)	400000	300010	300010	300010	300010	177490	177490	0
H ² CM	400000	300000	300000	300000	300000	166667	166667	0
Gain	0%	0%	0%	0%	0%	6%	6%	0%

Table 11. Comparaison des temps de calcul de planification.

Configurations testées		Temps de calcul (Sec)
Nombre de ressources composées	Nombre de Turbines	
2	6	0.14 x10 ⁻³
4	12	0.21 x10 ⁻³
6	18	0.28 x10 ⁻³
10	30	0.41 x10 ⁻³
20	60	0.74 x10 ⁻³
50	150	1.89 x10 ⁻³
100	300	3.92 x10 ⁻³
500	1500	16.96 x10 ⁻³
1000	3000	24.60x10 ⁻³

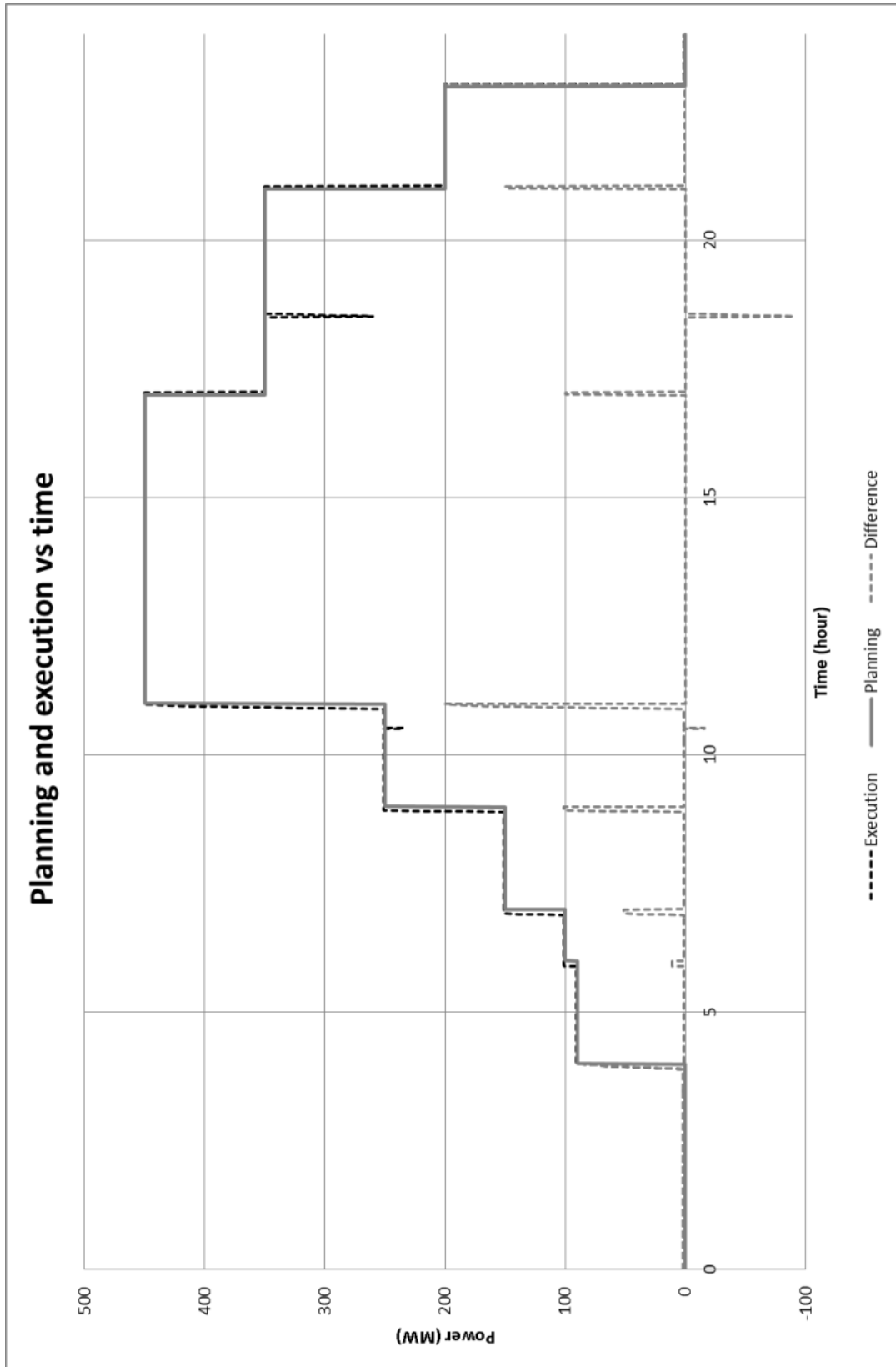


Figure 66. Planification et exécution de la production.

4.6 Revue du chapitre

Un deuxième exemple est présenté dans ce chapitre sur la base d'un système de production d'énergie utilisant une centrale à cycle combiné. Similairement au chapitre précédent, nous commençons par décrire l'exemple utilisé, suivi par l'explication de la procédure utilisée pour la simulation de l'exemple. Ensuite, la mise en œuvre du modèle de contrôle hybride holonique est détaillée, se terminant avec les résultats en termes de planification, de flexibilité et de performances du système.

Le principal intérêt de cet exemple est de montrer la simplicité de modélisation dans un cas industriel complexe. Le résultat immédiat est qu'il n'aurait pas été plus compliqué de modéliser un système de plus grande taille, ou de reconfigurer en ligne le système.

Conclusion et perspectives

Les travaux proposés dans cette thèse concernent l'implémentation d'une architecture holonique sur les systèmes de contrôle hybrides (HCS). Des propositions ont été faites pour adapter les éléments de base des HCS comme le contrôleur ou l'interface à la thématique générale des HMS. Le modèle proposé tend à confirmer la bonne adéquation d'une adaptation de l'architecture de référence classique PROSA à cette classe de systèmes, adaptation appelée H²CM. Ainsi, le contrôle orienté-événements est maintenu, mais avec la simplicité et la flexibilité intrinsèque de l'architecture holonique. De plus, une gestion plus simple du comportement des modèles, avec la possibilité de reconfigurer simplement le système de contrôle, a été observée.

H²CM est défini de telle façon que chaque holon Ressource a un holon Produit et un holon Ordre associé. Si le holon Ressource est composé, ses sous-holons Ressources seront également associés à un holon Produit et un holon Ordre, rendant le système fractal. L'utilisation d'un modèle fractal a l'avantage d'automatiser la génération du contrôle des ressources, car le modèle a un algorithme qui est indépendant du nombre de ressources et qui va toujours avoir la même structure, une caractéristique qui facilite la mise en œuvre de l'algorithme et donne de la flexibilité pour supprimer ou ajouter des ressources au HCS.

Les modèles de holons utilisés restent proches du modèle original de PROSA, avec quelques modifications. Le modèle du holon Ordre reste identique, mais le modèle du holon Produit et de holon Ressource ont une approche basée sur les services, cette caractéristique permettant le développement de produits indépendamment de la ressource qui fournit le service, et donne plus de flexibilité au système. Le modèle du holon Ressource composé va prendre des décisions selon les services fournis par les holons qui le composent, et il va négocier directement avec eux pendant la planification en donnant plus d'autonomie aux holons Ressources. Le holon Ressource a la structure Processus-Contrôle-Interface, qui permet à chaque holon d'être lui-même un système hybride.

L'exécution de H²CM est basée sur deux étapes : la planification et le suivi des ressources. Le processus de planification a été basé sur l'outil de négociation fréquemment utilisé sur les HMS mais nouveau pour le HCS. Le processus de planification en utilisant la négociation avec la structure hybride des holons fait que les modèles H²CM sont beaucoup plus faciles à développer, car au lieu de modéliser tout le système comme un ensemble indissociable comme il est traditionnellement fait dans HCS, il suffit juste de modéliser chaque holon individuel.

D'ailleurs, le fait que le système ait une structure fractale simplifie également le suivi de la production, car à chaque holon Ressource lui est associé un holon Ordre qui a entre autres fonctions de faire le suivi des objectifs du holon Ressource. Lorsqu'un holon Ressource est composé, alors l'état du holon Ressource dépend des états des ressources

qui le composent. Ainsi, la supervision de H^2CM est définie par le résultat de la négociation et de l'état du holon Ressource composé.

Comme indiqué à plusieurs reprises dans la littérature sur les systèmes holoniques, en plus de la caractéristique de flexibilité dont on a déjà parlé, les caractéristiques de réactivité, autonomie, et coopération sont des caractéristiques intrinsèques importantes de ceux-ci. H^2CM montre que les HCS peuvent acquérir ces caractéristiques. Pour illustrer cela, deux exemples ont été sélectionnés. Le premier, appelé système de commutation d'arrivée, qui est un exemple classique de système hybride, et le second, un exemple de process industriel basé sur une génération d'électricité dans une centrale à cycle combiné, ont été modélisés et leur pilotage via H^2CM a été implémenté et simulé sous plusieurs scénaris. Les résultats obtenus avec l'exemple de commutation d'arrivée montrent clairement que le système acquiert de la flexibilité et de la réactivité, car il est possible d'ajouter des réservoirs en ligne avec un contrôle qui continue à fonctionner sans plus de modification. Au travers de cet exemple, d'autres caractéristiques comme la performance et la stabilité du système sont évaluées. En étudiant ces résultats, nous avons remarqué que, même si des méthodes de contrôle analytiques peuvent les dépasser, l'implémentation de H^2CM permet d'obtenir une relativement bonne performance à la fois en termes de contrôle et de stabilité. Les délais de calcul de la planification sont de plus beaucoup plus courts que ceux des méthodes optimales testées à des fins comparatives dans cette thèse. Les délais de modélisation par H^2CM sont intuitivement plus courts grâce à la modélisation des ressources individuelles.

Les caractéristiques de coopération et d'autonomie sont observées sur les résultats du deuxième exemple. Ici, on introduit une série de défaillances ou de changements qui permettent de regarder l'évolution du comportement du système de contrôle. Lorsqu'un défaut ou une modification est faible, cela est résolu par un holon individuel, ce qui montre l'autonomie des holons. Quand une défaillance ou modification est de plus grande envergure, cela est résolu grâce à la coopération de plusieurs holons.

Nous proposons les perspectives suivantes concernant les développements à apporter à H^2CM :

L'utilisation d'autres familles de systèmes hybrides : Ce document a étudié une famille spécifique de systèmes hybrides appelés systèmes de contrôle hybride. Toutefois, la famille des systèmes hybrides comporte beaucoup d'éléments, il existe donc de nombreuses autres sous-familles comme par exemple les systèmes de contrôle commutés (Savkin and Evans, 2002) qui peuvent être également étudiés.

L'utilisation d'autres architectures de contrôle holonique : Le développement de H^2CM a été basé sur l'architecture holonique PROSA. De façon similaire aux familles de systèmes hybrides, il existe plusieurs autres architectures de référence holoniques ayant des caractéristiques différentes à celle de PROSA. Ces architectures ont des propriétés pouvant s'avérer très intéressantes, comme des nouveaux holons avec différentes répartitions des fonctionnalités ou de nouvelles procédures de planification ou de re-

planification. Une évolution de H²CM par rapport aux concepts proposés dans HCBA (Chirn and McFarlane, 2000) ou ADACOR (Leitão and Restivo, 2006) doit être menée afin d'améliorer H²CM, et essayer de créer un modèle généralisable améliorant les caractéristiques de l'architecture actuelle.

Amélioration les algorithmes de H²CM : Même si les algorithmes développés dans ce document ont un relativement bon fonctionnement sur les simulations réalisées, ils ont été créés de manière assez basique, sans idée d'optimiser leurs performances. L'idée est donc d'élaborer des codes de programmation plus génériques, avec des procédures plus robustes et moins sujettes à erreurs de programmation, améliorant aussi les méthodes utilisées dans les algorithmes au niveau de la supervision et de la négociation, incluant des protocoles de communication standardisés comme Contract-Net (Borangiu et al., 2014) ou un algorithme de suivi redondant.

Application sur d'autres bancs d'essai : Les exemples choisis sont représentatifs des systèmes hybrides, mais restent des simulations de process réels. La mise en œuvre sur systèmes réels apporte beaucoup de nouveaux facteurs pouvant modifier l'utilisation que l'on fait des variables. Une variable très importante de ce type de contrôle est le temps. Le résultat obtenu de la mise en œuvre de H²CM sur l'exemple de CCP montre le bon temps de traitement de l'algorithme, mais sur un process réel la contrainte temps-réel de prise de décision des équipements reste quelque chose qui n'a pas été pris en compte sur la simulation. Il serait donc intéressant de mettre en œuvre H²CM sur différents processus réels de façon à pouvoir faire une réelle estimation de performance de celui-ci.

Références

- Aissani, N., Beldjilali, B., Trentesaux, D., 2009. Dynamic scheduling of maintenance tasks in the petroleum industry: A reinforcement approach. *Eng. Appl. Artif. Intell.* 22, 1089–1103.
- Aissani, N., Beldjilali, B., Trentesaux, D., 2008. Use of machine learning for continuous improvement of the real time heterarchical manufacturing control system performances. *Int. J. Ind. Syst. Eng.* 3. doi:10.1504/IJISE.2008.017555
- Alla, H., David, R., 1998. Continuous and hybrid Petri nets. *J. Circuits Syst. Comput.* 8, 159–188.
- Alur, R., Courcoubetis, C., Henzinger, T.A., Ho, P.-H., 1993. *Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems.* Springer.
- ANSI/ISA-95, 2010. ANSI/ISA-95.00.01-2010 (IEC 62264-1 Mod) Enterprise-Control System Integration - Part 1: Models and Terminology.
- Antsaklis, P.J., 1998. Hybrid Control Systems: An Introductory Discussion to the Special Issue. *IEEE Trans. Autom. Control* 43, 457–460. doi:10.1109/TAC.1998.664148
- Antsaklis, P.J., Passino, K.M., 1993. *Introduction to intelligent control systems with high degrees of autonomy.* Kluwer Academic Publishers.
- Antsaklis, P.J., Stiver, J.A., Lemmon, M., 1993. Hybrid system modeling and autonomous control systems, in: Grossman, R.L., Nerode, A., Ravn, A.P., Rischel, H. (Eds.), *Hybrid Systems, Lecture Notes in Computer Science.* Springer Berlin Heidelberg, pp. 366–392.
- Ashby, W.R., 1956. *An introduction to cybernetics,* MBLWHOI Library. ed. New York, J. Wiley.
- Aytug, H., Lawley, M.A., McKay, K., Mohan, S., Uzsoy, R., 2005. Executing production schedules in the face of uncertainties: A review and some future directions. *Eur. J. Oper. Res., IEPM: Focus on Scheduling* 161, 86–110. doi:10.1016/j.ejor.2003.08.027
- Barbosa, J., Leitão, P., Adam, E., Trentesaux, D., 2015. Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution. *Comput. Ind.* 66, 99–111. doi:10.1016/j.compind.2014.10.011
- Bemporad, A., Morari, M., 1999. Control of systems integrating logic, dynamics, and constraints. *Automatica* 35, 407–427.
- Berger, T., Sallez, Y., Valli, B., Gibaud, A., Trentesaux, D., 2010. Semi-heterarchical Allocation and Routing Processes in FMS Control: A Stigmergic Approach. *J. Intell. Robot. Syst.* 58, 17–45. doi:10.1007/s10846-009-9343-9
- Blanc, P., Demongodin, I., Castagna, P., 2008. A holonic approach for manufacturing execution system design: An industrial application. *Eng. Appl. Artif. Intell.* 21, 315–330.
- Borangui, T., Raileanu, S., Trentesaux, D., Berger, T., 2010. Open manufacturing control with agile reconfiguring of resource services. *J. Control Eng. Appl. Inform.* 12, 10–17.
- Borangui, T., Raileanu, S., Trentesaux, D., Berger, T., Iacob, I., 2014. Distributed manufacturing control with extended CNP interaction of intelligent products. *J. Intell. Manuf.* 25, 1065–1075. doi:10.1007/s10845-013-0740-3
- Branicky, M.S., 1995. *Studies in hybrid systems: Modeling, analysis, and control.* DTIC Document.

- Cardin, O., Castagna, P., 2011. Proactive production activity control by online simulation. *Int. J. Simul. Process Model.* 6, 177–186.
- Cardin, O., Castagna, P., 2009. Using online simulation in Holonic Manufacturing Systems. *Eng. Appl. Artif. Intell.* 22, 1025–1033.
- Cardin, O., Castagna, P., Chové, E., 2009. Utilisation d'un observateur pour le pilotage des systèmes contrôlés par le produit. *J. Eur. Systèmes Autom.* 43, 463–486. doi:doi:10.3166/jesa.43.463-486
- Cardin, O., Mebarki, N., Pinot, G., 2013. A study of the robustness of the group scheduling method using an emulation of a complex FMS. *Int. J. Prod. Econ.* 146, 199–207. doi:10.1016/j.ijpe.2013.06.023
- Cassandras, C.G., Lafortune, S., 2008. *Introduction to Discrete Event Systems*. Springer.
- Chacón, E., Besembel, I., Narciso, F., Montilva, J., Colina, E., 2002. An integration architecture for the automation of a continuous production complex. *ISA Trans.* 41, 95–113. doi:10.1016/S0019-0578(07)60205-5
- Chase, C., Serrano, J., Ramadge, P.J., 1993. Periodicity and chaos from switched flow systems: contrasting examples of discretely controlled continuous systems. *IEEE Trans. Autom. Control* 38, 70–83. doi:10.1109/9.186313
- Che, A., Zhou, Z., Chu, C., Chen, H., 2011. Multi-degree cyclic hoist scheduling with time window constraints. *Int. J. Prod. Res.* 49, 5679–5693. doi:10.1080/00207543.2010.503200
- Chirn, J.-L., McFarlane, D.C., 2000. A holonic component-based approach to reconfigurable manufacturing control architecture, in: *11th International Workshop on Database and Expert Systems Applications Proceedings*. Presented at the 11th International Workshop on Database and Expert Systems Applications, 2000. Proceedings, pp. 219–223. doi:10.1109/DEXA.2000.875030
- Chokshi, N., McFarlane, D., 2008. A distributed architecture for reconfigurable control of continuous process operations. *J. Intell. Manuf.* 19, 215–232.
- Chokshi, N., McFarlane, D., 2007. *A Distributed Coordination Approach to Reconfigurable Process Control*. Springer.
- Chové, E., Castagna, P., Abbou, R., 2009. Hoist Scheduling Problem: Coupling reactive and predictive approaches, in: *INCOM 2009*. pp. 2077–2082.
- David, R., Alla, H., 2010. *Discrete, continuous, and hybrid Petri nets*. Springer.
- De Mello, F.P., Ahner, D.J., Anderson, P.M., Doudna, J.H., Kundur, P., Richardson, L.M., Tandy, G., Taylor, C.W., Van De Meulebroeke, F., 1994. Dynamics models for combines cycle plants in power system studies, in: *IEEE Transactions on Power Systems*. Presented at the IEEE/PES 1994 Winter meeting, Institute of Electrical and Electronics Engineers, pp. 1698–1708.
- Demongodin, I., Koussoulas, N.T., 1998. Differential Petri nets: Representing continuous systems in a discrete-event world. *IEEE Trans. Autom. Control* 43, 573–579.
- Dubromelle, Y., Prunaret, J.P., Ounnar, F., Pujo, P., 2014. RFID synchronisation for ambient services in a holonic and isoarchic control model. *Int. J. Radio Freq. Identif. Technol. Appl.* 4, 272. doi:10.1504/IJRFITA.2014.063925
- Gamboa Quintanilla, F., Cardin, O., Castagna, P., 2014. Product Specification for Flexible Workflow Orchestrations in Service Oriented Holonic Manufacturing Systems, in: *Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics, Studies in Computational Intelligence*. Springer, pp. 177–193.
- Gamboa Quintanilla, F., Cardin, O., Castagna, P., 2013. Evolution of a Flexible Manufacturing System: From Communicating to Autonomous Product, in: *Service Orientation in Holonic and Multi Agent Manufacturing and Robotics, Studies in Computational Intelligence*. Springer, pp. 167–180.

- Gouyon, D., Pétin, J.-F., Morel, G., 2007. A product-driven reconfigurable control for shop floor systems. *Stud. Inform. Control* 16.
- Henzinger, T.A., 2000. *The theory of hybrid automata*. Springer.
- Hilario, M., Pellegrini, C., Alexandre, F., 1994. Modular integration of connectionist and symbolic processing in knowledge-based systems. Presented at the International Symposium on Integrating Knowledge and Neural Heuristics, Pensacola, Italy, pp. 123–132.
- Iglesias, C.A., González, J.C., Velasco, J.R., 1996. MIX: A general purpose multiagent architecture, in: *Intelligent Agents II Agent Theories, Architectures, and Languages*. Springer, pp. 251–266.
- Indriago, C., Cardin, O., Morineau, O., Rakoto, N., Castagna, P., 2015a. Performance evaluation of holonic-based online predictive-reactive scheduling for a switch arrival system, in: *Proceedings of INCOM 2015*. Ottawa, Canada.
- Indriago, C., Cardin, O., Rakoto, N., Chacón, E., Castagna, P., 2015b. Application of Holonic Paradigm to Hybrid Processes: Case of a Water Treatment Process, in: *Service Orientation in Holonic and Multi-Agent Manufacturing, Studies in Computational Intelligence*. Springer, pp. 39–48.
- Indriago, C., Cardin, O., Rakoto-Ravalontsalama, N., Chacón, E., Castagna, P., 2014a. Application du paradigme holonique à un système de réservoirs, in: *10ème Conférence Francophone de Modélisation, Optimisation et Simulation-MOSIM'14*. Presented at the 10ème Conférence Francophone de Modélisation, Optimisation et Simulation- MOSIM'14, Nancy –France.
- Indriago, C., Cardin, O., Rakoto-Ravalontsalama, N., Chacón, E., Castagna, P., 2014b. Application of holonic paradigm to hybrid processes: case of a water treatment process, in: *SOHOMA'14 Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*. Presented at the SOHOMA'14 Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing, Nancy –France.
- Koestler, A., 1990. *The Ghost in the Machine*. Penguin Books, London.
- Kohn, W., Nerode, A., 1992. Multiple agent autonomous hybrid control systems, in: *Proceedings of the 31st IEEE Conference on Decision and Control*, 1992. Presented at the *Proceedings of the 31st IEEE Conference on Decision and Control*, 1992, pp. 2956–2966 vol.3. doi:10.1109/CDC.1992.371270
- Koutsoukos, X.D., Antsaklis, P.J., Stiver, J.A., Lemmon, M.D., 2000. Supervisory control of hybrid systems. *Proc. IEEE* 88, 1026–1049. doi:10.1109/5.871307
- Labinaz, G., Bayoumi, M.M., Rudie, K., 1997. A survey of modeling and control of hybrid systems. *Annu. Rev. Control* 21, 79–92. doi:10.1016/S1367-5788(97)00019-9
- Labinaz, G., Rudie, K., Ricker, L., Sarkar, N., Bayoumi, M.M., 1996. A hybrid system investigation of fluid-filled tanks. *Summ. This Work Was Present. Math. Theory Netw. Syst.* 96–01.
- Langer, G., 1999. *HoMuCS-A methodology and architecture for Holonic Multi-cell Control Systems*. IPT, DTU.
- Lee, E.A., 2006. Cyber-Physical Systems - Are Computing Foundations Adequate?, in: *Position Paper for NSF Workshop On Cyber-Physical Systems*. Austin, Texas.
- Leitão, P., Restivo, F., 2006. ADACOR: A holonic architecture for agile and adaptive manufacturing control. *Comput. Ind.* 57, 121–130. doi:10.1016/j.compind.2005.05.005
- Le Mortellec, A., Clarhaut, J., Sallez, Y., Berger, T., Trentesaux, D., 2013. Embedded holonic fault diagnosis of complex transportation systems. *Eng. Appl. Artif. Intell.* 26, 227–240. doi:10.1016/j.engappai.2012.09.008

- Manier, M.-A., Bloch, C., 2003. A Classification for Hoist Scheduling Problems. *Int. J. Flex. Manuf. Syst.* 15, 37–55. doi:10.1023/A:1023952906934
- McFarlane, D.C., 1995. Holonic manufacturing systems in continuous processing: concepts and control requirements, in: *Proceedings of ASI*. Citeseer, p. 282a273.
- Mehta, S.V., Uzsoy, R.M., 1998. Predictable scheduling of a job shop subject to breakdowns. *IEEE Trans. Robot. Autom.* 14, 365–378. doi:10.1109/70.678447
- Mesarovic, M.D., Macko, D., Takahara, Y., 1970. Theory of hierarchical, multilevel systems. 1970. Mesarovic34Theory Hierarchical Multilevel Syst. 34.
- Mettala, E., Graham, M.H., 1992. The domain-specific software architecture program. DTIC Document.
- Monostori, L., 2014. Cyber-physical Production Systems: Roots, Expectations and R&D Challenges. *Procedia CIRP, Variety Management in Manufacturing Proceedings of the 47th CIRP Conference on Manufacturing Systems 17*, 9–13. doi:10.1016/j.procir.2014.03.115
- Moor, T., Davoren, J.M., Anderson, B.D.O., 2002. Robust hybrid control from a behavioural perspective, in: *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*. IEEE, pp. 1169–1174.
- Mosterman, P.J., 1997. Hybrid Dynamic Systems: A hybrid bond graph modeling paradigm and its application in diagnosis. Vanderbilt University.
- Nerode, A., Kohn, W., 1993a. Models for hybrid systems: Automata, topologies, controllability, observability, in: Grossman, R.L., Nerode, A., Ravn, A.P., Rischel, H. (Eds.), *Hybrid Systems, Lecture Notes in Computer Science*. Springer Berlin Heidelberg, pp. 317–356.
- Nerode, A., Kohn, W., 1993b. Multiple agent hybrid control architecture. Springer.
- Nerode, A., Kohn, W., 1992. An autonomous systems control theory: An overview, in: *Computer-Aided Control System Design, 1992.(CACSD), 1992 IEEE Symposium on*. IEEE, pp. 204–210.
- Nicollin, X., Olivero, A., Sifakis, J., Yovine, S., 1993. An approach to the description and analysis of hybrid systems, in: *Hybrid Systems*. Springer, pp. 149–178.
- Pach, C., Berger, T., Bonte, T., Trentesaux, D., 2014. ORCA-FMS: a dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling. *Comput. Ind.* 65, 706–720. doi:10.1016/j.compind.2014.02.005
- Pannequin, R., Morel, G., Thomas, A., 2009. The performance of product-driven manufacturing control: An emulation-based benchmarking study. *Comput. Ind.* 60, 195–203. doi:10.1016/j.compind.2008.12.007
- Paynter, H.M., 1961. *Analysis and design of engineering systems*. MIT press.
- Petri, C.A., 1966. *Communication with automata*.
- Polderman, J.W., Willems, J.C., 2000. *Introduction to mathematical systems theory*.
- Pujo, P., Broissin, N., Ounnar, F., 2009. PROSIS: An isoarchic structure for HMS control. *Eng. Appl. Artif. Intell., Distributed Control of Production Systems 22*, 1034–1045. doi:10.1016/j.engappai.2009.01.011
- Rowen, W.I., 1992. Simplified Mathematical Representations of Single Shaft Gas Turbines in Mechanical Drive Service, in: *ASME Proceedings. Presented at the International Gas Turbine and Aeroengine Congress and Exposition*. doi:10.1115/92-GT-022
- Rowen, W.I., 1983. Simplified Mathematical Representations of Heavy-Duty Gas Turbines. *J. Eng. Gas Turbines Power* 105, 865–869. doi:10.1115/1.3227494
- Sallez, Y., 2014. Proposition of an Analysis Framework to Describe the “Activeness” of a Product during Its Life Cycle, in: *Service Orientation in Holonic and Multi-Agent*

- Manufacturing and Robotics, Studies in Computational Intelligence. Springer, pp. 257–270.
- Sallez, Y., Berger, T., Raileanu, S., Chaabane, S., Trentesaux, D., 2010. Semi-heterarchical control of FMS: From theory to application. *Eng. Appl. Artif. Intell.* 23, 1314–1326. doi:10.1016/j.engappai.2010.06.013
- Sallez, Y., Berger, T., Trentesaux, D., 2009. Open-Control: A New Concept for Integrated Product-Driven Manufacturing Control, in: Bakhtadze, N. (Ed.), *INCOM 2009*. pp. 2065–2070. doi:10.3182/20090603-3-RU-2001.00347
- Savkin, A.V., Evans, R.J., 2002. *Hybrid dynamical systems: controller and sensor switching problems*. Springer Science & Business Media.
- Scheer, A.-W., 2012. *CIM. Computer Integrated Manufacturing: Towards the Factory of the Future*. Springer Science & Business Media.
- Serugendo, G.D.M., Karageorgos, A., Rana, O.F., Zambonelli, F., 2004. *Engineering Self-Organising Systems: Nature-Inspired Approaches to Software Engineering*. Springer.
- Sontag, E.D., 1998. *Mathematical control theory*, volume 6 of. *Texts Appl. Math.* 54.
- Suda, H., 1989. Future factory system formulated in Japan. *Techno Jpn.* 22, 15–25.
- Tavakoli, M.R.B., Vahidi, B., Gawlik, W., 2009. An Educational Guide to Extract the Parameters of Heavy Duty Gas Turbines Model in Dynamic Studies Based on Operational Data. *IEEE Trans. Power Syst.* 24, 1366–1374. doi:10.1109/TPWRS.2009.2021231
- Tian, Y.-P., 2005. Delayed feedback control of chaos in a switched arrival system. *Phys. Lett. A* 339, 446–454. doi:10.1016/j.physleta.2005.03.061
- Trivedi, K.S., Kulkarni, V.G., 1993. FSPNs: fluid stochastic Petri nets, in: *Application and Theory of Petri Nets 1993*. Springer, pp. 24–31.
- Ushio, T., Ueda, H., Hirai, K., 1995. Controlling chaos in a switched arrival system. *Syst. Control Lett.* 26, 335–339. doi:10.1016/0167-6911(95)00032-1
- Valckenaers, P., Van Brussel, H., Wyns, J., Bongaerts, L., Peeters, P., 1998. Designing Holonic manufacturing systems. *Robot. Comput.-Integr. Manuf.* 14, 455–464. doi:10.1016/S0736-5845(98)00020-9
- Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P., 1998. Reference architecture for holonic manufacturing systems: PROSA. *Comput. Ind.* 37, 255–274. doi:10.1016/S0166-3615(98)00102-X
- Van Der Schaft, A.J., Schumacher, J.M., van der Schaft, A.J., van der Schaft, A.J., 2000. *An introduction to hybrid dynamical systems*. Springer London.
- Verstraete, P., Saint Germain, B., Valckenaers, P., Van Brussel, H., Van Belle, J., Karuna, H., 2008. Engineering Manufacturing Control Systems Using PROSA and Delegate MAS. *Int J Agent-Oriented Softw Eng* 2, 62–89. doi:10.1504/IJAOSE.2008.016800
- Villa Ramirez, J.L., 2004. *Modélisation et commande de systèmes hybrides: l’approche MLD*. Nantes.
- Von Bertalanffy, L., 1973. *General System Theory: Foundations, Development, Applications*: Rev. Ed. George Braziller.
- Wong, C.Y., McFarlane, D., Ahmad Zaharudin, A., Agarwal, V., 2002. The intelligent product driven supply chain, in: *2002 IEEE International Conference on Systems, Man and Cybernetics*. Presented at the 2002 IEEE International Conference on Systems, Man and Cybernetics, p. 6 pp. vol.4–. doi:10.1109/ICSMC.2002.1173319

Références

- Yang, X., Moore, P., Chong, S.K., 2009. Intelligent products: From lifecycle data acquisition to enabling product-related services. *Comput. Ind.* 60, 184–194. doi:10.1016/j.compind.2008.12.009
- Zapata Madrigal, G.D., 2011. Propuesta Para la Planificación, Programación, Supervisión y Control de la Producción en Procesos Continuos Desde la Teoría del Control Supervisorio y el Enfoque Holónico. Universidad de Los Andes, Merida, Venezuela.

Thèse de Doctorat

Carlos INDRIAGO

APPLICATION DU PARADIGME HOLONIQUE AUX SYSTEMES DE CONTRÔLE HYBRIDES

Résumé

De nombreux travaux s'intéressent à l'interaction entre des processus de fabrication continue et des algorithmes de planification discrets, ce qui a créé une classe de systèmes hybrides appelées « Hybrid Control Systems » (HCS). Ces systèmes peuvent être décomposés en trois niveaux distincts : les niveaux contrôleur, atelier et interface. Les HCS sont traditionnellement modélisés par des systèmes à commutation représentant le système complet plutôt que des ressources atomiques. De ce fait, les processus de reconfiguration sont complexes car il est nécessaire de considérer le système dans sa globalité, rendant les HCS peu flexibles et plus sensibles aux incertitudes. Ce besoin de flexibilité a donc poussé plusieurs équipes à travailler sur l'application du paradigme holonique aux HCS.

L'objectif de cette thèse est de démontrer la possibilité d'appliquer pratiquement directement l'une des architectures de référence holonique des systèmes à événements discrets aux HCS. Pour ce faire, nous proposons une nouvelle architecture appelée Holonic Hybrid Control Model (H^2CM), basée sur l'architecture PROSA, et son application sur les HCS.

Deux cas d'études sont présentés pour vérifier le fonctionnement correct du modèle proposé ; le premier est un exemple typique de la littérature des systèmes hybrides appelé Système à Commutations d'Arrivée, tandis que le second est inspiré d'un processus industriel de génération d'électricité, spécifiquement un Processus à Cycles Combinés. Ces deux cas d'études ont été simulés et un ensemble d'expérimentations ont été menées afin d'évaluer les caractéristiques de flexibilité, autonomie, coopération, performance et stabilité du système de contrôle obtenu.

Mots clés: Hybrid Control Systems, architectures de référence holonique, Holonic Hybrid Control Model, Système à Commutations d'Arrivée, Processus à Cycles Combinés.

Abstract

Several studies typically arise from the interaction of discrete planning algorithms and continuous processes, normally called hybrid control systems. It consists in three distinct levels, the controller, the plant and the interface. Hybrid control systems are conventionally modeled by switching patterns using the whole system instead of atomic resource. Therefore, the reconfiguration process is complex because it must take into account the system as a whole, making the hybrid control systems inflexible and more susceptible to uncertainties. The need for flexibility thus leads several teams to investigate the application of holonic paradigm to hybrid control systems.

The objective of this document is to demonstrate the possibility to apply almost directly a holonic discrete-event based reference architecture to hybrid control systems. To do this, it is proposed a model called holonic hybrid control model (H^2CM) derived from the PROSA architecture application on hybrid control systems.

Two study cases were taken for verifying the proper operation of the proposed model; the first is a typical example from the hybrid systems literature called switched arrival systems and a second is an example from a real process in industrial electricity generation process, specifically a Combined Cycle Plant (CCP). Both study cases were simulated in Java and a series of experiments looking for measure flexibility, autonomy, cooperativeness, performance and stability were applied.

Key words: hybrid control systems, holonic manufacturing systems, holonic hybrid control model, switched arrival systems, combined cycle plant.