

APPLICATION DU PARADIGME HOLONIQUE A UN SYSTEME DE RESERVOIRS

C. INDRIAGO, O. CARDIN, N. RAKOTO,
P. CASTAGNA

LUNAM Université, IUT de Nantes – Université de
Nantes, IRCCyN UMR CNRS 6597,
2 avenue du Prof. Jean Rouxel – 44475 Carquefou
Carlos.Indriago, Olivier.Cardin, Pierre.Castagna@univ-
nantes.fr, Naly.Rakoto@mines-nantes.fr

E. CHACON

Universidad de los Andes
La Hechicera Mérida 5101 Venezuela
echacon@ula.ve

RESUME : *Le paradigme holonique a été largement étudié dans le cadre de la production manufacturière. Ces productions sont de type discrètes, car les opérations incluses dans les gammes ne mettent pas en jeu de variables à évolution continue. Une classe différente de systèmes est étudiée ici, les systèmes hybrides, qui englobe l'ensemble des systèmes ayant une évolution continue par morceaux, et dont les changements sont liés à l'évolution de variables discrètes. Dans ce cas, une reconfiguration du système est généralement nécessaire, et le paradigme holonique est une réponse pertinente au besoin de flexibilité en découlant. Ce papier propose l'application d'une architecture de référence connue de la littérature discrète au cas des systèmes hybrides. Le modèle proposé est composé de l'union du modèle hybride et du modèle holonique, où le contrôleur proposé est représenté à la fois par l'holon Produit et l'holon Ordre, l'interface est représentée par la partie logique du holon Ressources et le système de production est représenté par la partie physique du holon ressource. La spécification du holon produit utilisé est orientée services, et la spécification du holon ressource est une abstraction de la partie physique de la ressource contenant tous les modèles de conversion d'état discret vers l'état continu et vice-versa. Un cas d'étude s'appuyant sur un système de commutations d'arrivées a été choisi et illustre la pertinence de l'architecture de référence discrète pour modéliser les systèmes hybrides.*

MOTS-CLES : *Systèmes hybrides, systèmes holoniques, système de réservoirs, PROSA.*

1 INTRODUCTION

Dans les systèmes de production, il y a les systèmes de production discrète où les différents produits sont réalisés en utilisant diverses opérations discrètes telles que l'usinage, le perçage, le meulage, etc. Ces pièces sont ensuite associées sur une ligne d'assemblage pour créer le produit fini. D'un autre côté, il y a les processus continus qui impliquent des flux continus de matériaux (comme des liquides, des poudres ou de l'énergie) et des services à travers des unités de traitement interconnectées via des tuyaux ou des câbles.

Ainsi, entre les systèmes discrets et continus il y a des systèmes que (Chokshi and McFarlane, 2007) appellent processus semi-continus, qui, sont similaires à des processus continus impliquant des flux continus de matériaux et de services, mais dont l'état du système peut être discret, dans le sens où sa dynamique change à des instants précis correspondant à des événements.

Les processus semi-continus sont particulièrement importants dans ce travail, particulièrement les systèmes appelés « systèmes hybrides (SH) », qui sont classiquement structurés en trois différents niveaux (Koutsoukos et al., 2000), illustrés par la figure 1. Le contrôleur est considéré comme un système à événements discrets (SED). Il reçoit, traite et génère des événements qui sont représentés par des variables discrètes. Le système de

production (SP) est un système continu généralement modélisés par des équations différentielles. Le SP reçoit des signaux, traite et génère des sorties qui sont représentés par les variables réelles qui sont généralement continues. Le contrôleur et le SP communiquent via une interface qui traduit les sorties du SP en variables discrètes pour le contrôleur, et les variables de sortie du contrôleur en signaux de commande pour l'entrée du SP. L'interface peut être considérée comme constituée de deux sous-systèmes: le générateur, qui détecte les sorties du SP et génère des variables discrètes représentant des événements du SP, et l'actionneur, qui traduit les ordres discrets générés par le contrôleur en signaux d'entrée du SP.

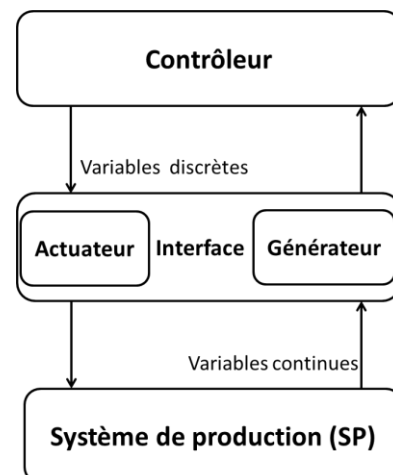


Figure 1. Architecture des systèmes hybrides

Les systèmes hybrides sont classiquement modélisés par des modèles de commutation concernant l'ensemble du système. A cause de cela, tout processus de reconfiguration est complexe parce qu'il doit prendre en compte ce système dans son ensemble. D'où la nécessité de trouver des nouvelles architectures pour ces systèmes hybrides permettant d'améliorer leur flexibilité en facilitant leur reconfiguration.

Les architectures holoniques ((Van Brussel et al., 1998), (Leitão, and Restivo 2006) ou encore (Chirn and McFarlane, 2000)) sont des architectures flexibles permettant une reconfiguration des systèmes et des procédés pour la création de nouveaux produits. Une architecture de fabrication holonique doit faciliter l'(auto-) configuration, l'extension et les modifications du système, en permettant une plus grande flexibilité en augmentant l'espace de décision aux niveaux supérieurs. La structure de l'architecture de référence PROSA (Van Brussel et al., 1998) est construite autour de trois types d'holons de base: holon Ordre, holon Produits, et holon Ressources. Chacun d'eux est responsable d'un aspect pilotage de la fabrication, que ce soit au niveau logistique, au niveau planification, ou au niveau des ressources. Parmi l'ensemble des architectures de référence holoniques pouvant être trouvées dans la littérature, PROSA a la particularité de proposer des holons ayant un niveau conceptuel relativement haut et, tout en étant dédié aux systèmes de production en général, indépendants de l'application visée. Pour ces raisons cette architecture a été choisie comme étant la première à être appliquée à un système hybride.

Plusieurs travaux proposent déjà d'utiliser des modèles holoniques pour des systèmes continus (McFarlane, 1995), (Chacón et al., 2002), (Chokshi and McFarlane, 2008) Une synthèse de ces travaux présentée en section 2 montre que les auteurs ont développé des holarchies très spécifiques, conçues empiriquement à partir des fonctionnalités attendues du système, sans proposer l'implémentation d'holarchies classiques. L'objectif principal de ce travail est de démontrer que les architectures de référence holoniques, conçues pour fonctionner sur des systèmes discrets, sont également performantes pour l'implémentation de systèmes hybrides.

La proposition de cet article est de montrer l'application de l'architecture PROSA sur un système hybride et vérifier son bon fonctionnement. Le cas d'étude proposé est très simple et fait partie des cas les plus utilisés de la communauté des systèmes hybrides : «système de commutation d'arrivées», proposé initialement par (Chase et al., 1993) et présenté dans la section 3. L'objectif est de montrer les bases de la modélisation du système hybride selon les trois holons de base de PROSA, sections 4 et 5.

Enfin, la section 6 propose une simulation de cet exemple en Java et les résultats qui montrent le potentiel

de l'architecture utilisée pour le contrôle et la reconfiguration du système.

2 ETAT DE L'ART

L'application des systèmes holoniques sur des systèmes continus a déjà été étudiée par quelques auteurs. Un cadre de contrôle pour systèmes hybrides (Figure 2) est défini par (McFarlane, 1995). L'objectif du système de contrôle a la particularité de convertir en interne les objectifs ou les exigences en un ensemble de paramètres ou de trajectoires de comportement admissibles, via une fonction de décision donnée, D . L'action de contrôle effective est alors réalisée sur le processus P . La terminologie utilisée dans la figure 2 est la suivante:

- m : ensemble des décisions de remplacement
- u : ensemble d'événements alternatifs
- y : ensemble des résultats possibles

Où $m \in M, u \in U, y \in Y, P : M \times U \rightarrow Y$

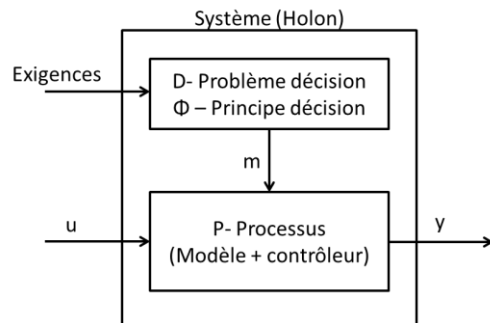


Figure 2. Cadre de contrôle pour SH

Le développement de problèmes de décision appropriés est la clé de l'intégration de l'activité de contrôle au sein des systèmes holoniques. Les instances de ces holons peuvent avoir des objectifs potentiellement différents selon l'environnement dans lequel ils opèrent. Des mécanismes de prise de décision basés sur l'optimisation distribuée sont très appropriés pour les systèmes holoniques et des méthodes de référencement pourraient aider à l'intégration des algorithmes de contrôle basés sur l'optimisation des opérations de fabrication.

Une autre architecture holonique développée pour les systèmes continus est celle proposée par (Chacón et al., 2002). Il est proposé une structure holonique basée sur un système de prise de décision appelé unité de production (Figure 3). Cette unité est la base pour une structure exploitant la caractéristique fractale des systèmes holoniques. L'agrégation des unités de production est alors appelé système de production complexe.

L'unité de production a des entrées et des sorties physiques (matières premières, produits finis, services) et logiques (objectifs, bulletins d'information). De plus, cette unité de production suit une structure holonique composée de trois holons de bases appelés holon ressource, holon mission-produit et holon ingénierie.

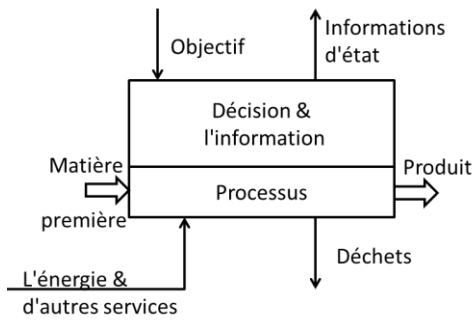


Figure 3. Unité de production

On peut enfin mentionner l'ouvrage de (Chokshi and McFarlane, 2007) qui développe une approche de coordination distribuée pour des processus de contrôle reconfigurables. Dans ce travail, une architecture de contrôle est développée en utilisant des éléments de base comme l'élément produit, l'élément unité, l'élément d'en-tête et l'élément de service. La Figure 4 montre ces éléments et comment ils interagissent. Même si elle ne traite pas explicitement du paradigme holonique, l'objectif de cette architecture est d'avoir un système de contrôle basé sur des modèles d'interaction reconfigurables et flexibles, semblables aux systèmes holoniques, et qui peuvent être implémentés dans des procédés continus et semi-continus.

Plusieurs approches ont donc été appliquées pour le contrôle de systèmes hybrides. Ces approches étant systématiquement ad hoc, elles peuvent difficilement bénéficier des nombreux résultats obtenus sur les systèmes discrets, ce qui serait facilité en appliquant les architectures de référence classiques. L'objectif des sections suivantes est d'étudier l'opportunité d'appliquer une architecture discrète classique à un système hybride, en commençant par la description du système choisi.

3 CAS D'ETUDE

Cette section décrit un exemple de système hybride, appelé système de commutations d'arrivée (de flux), et présenté initialement dans (Chase et al., 1993). Plusieurs auteurs ont utilisé ce système comme illustration de travaux divers, notamment sur la détection de défaut-

lance, 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 Présentation du système

Le système est constitué de n réservoirs et un serveur. Le fluide est retiré du réservoir 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 5 montre le système avec $n = 3$ et le serveur localisé sur le réservoir 1.

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.

Soit $x_i(t)$ la quantité de liquide dans un réservoir i au temps $t > 0$, et soit $x(t) = (x_1(t), \dots, x_n(t))$. A $t=0$, nous supposons que $x_i(0) \geq 0$ avec $\sum_{i=1}^n x_i(0) = h$. Nous appelons $x(t)$ l'état des réservoirs à l'instant t . 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 :

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

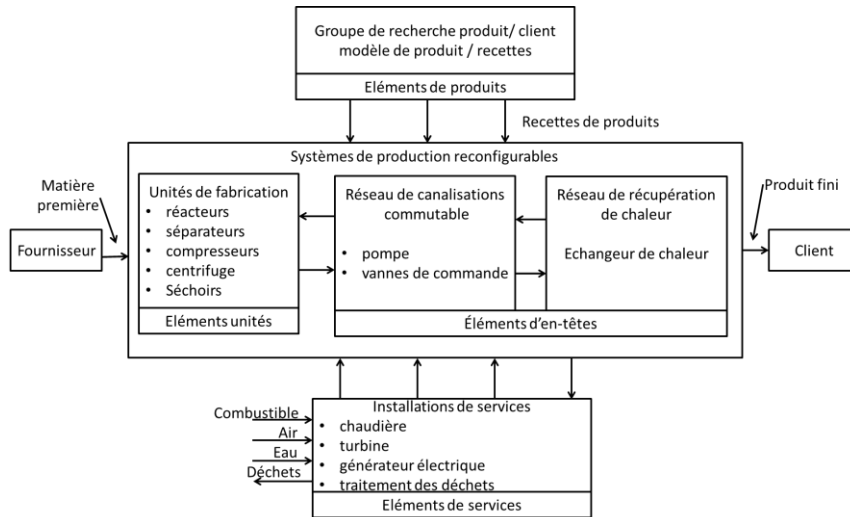


Figure 4. Éléments du processus de base dans un contrôle de processus reconfigurable

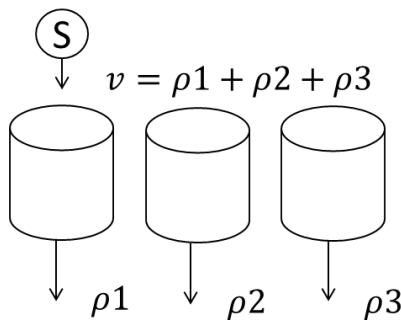


Figure 5. Système de commutations d'arrivée avec n=3.

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.

3.2 Problème de pilotage

La séquence d'états de niveaux $\{x(n)\}$ appartient à l'ensemble. $X = \{x: \sum_i x_i = h, x_i \geq 0, \text{ et pour certains } j, x_j = 0\}$. Si $G_j: X \rightarrow X$ désigne la fonction qui décrit la transformation de X qui résulte en plaçant le serveur à l'emplacement j jusqu'à ce qu'un autre réservoir se vide, alors

$$G_j(x) = x + \left(\min_{k \neq j} \left\{ \frac{x_k}{\rho_k} \right\} \right) (v_j - \rho)$$

Où ρ est le vecteur des ρ_i , et v_j est le vecteur de zéros à l'exception d'une valeur v dans la j -ième position. Pour $n=3$ le comportement de G est représenté par la figure 6, où il y a un point p_1 de coordonnées $\left(0, \frac{h \cdot \rho_2}{\rho_2 + \rho_3}, \frac{h \cdot \rho_3}{\rho_2 + \rho_3}\right)$ et si $\begin{pmatrix} x_2 \\ \rho_2 \end{pmatrix} = \begin{pmatrix} x_3 \\ \rho_3 \end{pmatrix}$ alors $G = (h, 0, 0)$ où deux réservoirs sont vides en même temps.

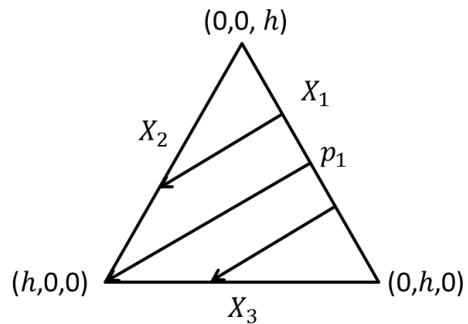


Figure 6. Comportement de G pour $n=3$.

Quand deux réservoirs sont vides en même temps, le système devient non déterministe et le contrôle classique ne donne aucune solution.

3.3 Solution développée en hybride

Pour trouver une solution, les systèmes hybrides utilisent le modèle du système entier. 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, 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.

4 METHODOLOGIE

Les caractéristiques des systèmes holoniques comme la flexibilité, la reconfiguration et l'autodiagnostic sont des caractéristiques souhaitables sur les systèmes hybrides. L'objectif de ce papier est de montrer comment s'adapte une architecture holonique classique comme PROSA, initialement développée pour les systèmes discrets, sur un système hybride. Pour cela, les sections suivantes définissent l'adaptation des holons Ordre, Produit et Ressource pour les systèmes hybrides.

4.1 Modélisation holonique du système

Une architecture hybride est composée de trois parties, comme mentionné dans les paragraphes précédents : le contrôleur, l'interface et le système de production. Dans le contrôleur du système hybride proposé le modèle est représenté par les holons Produit et Ordre, l'interface est représentée par la partie logique du holon Ressource et le système de production est représenté par la partie physique du holon Ressource. La figure 7 illustre le passage de la décomposition classique des systèmes hybrides en une modélisation suivant le paradigme holonique.

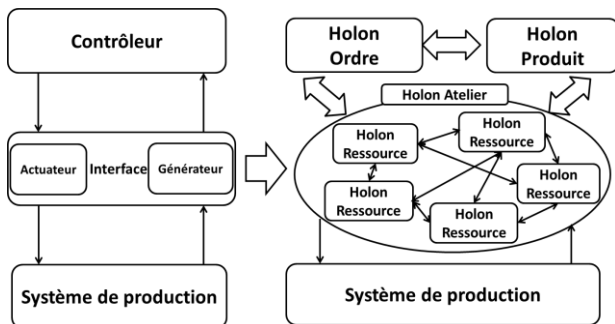


Figure 7. Architecture holonique d'un système hybride.

Ainsi, les données qui transitent depuis le système de production vers la partie logique du holon ressource et vice versa sont des données continues, tandis que les données vers le holon Ordre depuis la partie logique du holon Ressource sont des données discrètes. Cette caractéristique permet à l'holon Ordre d'avoir des caractéristiques presque invariables par rapport à une utilisation classique de PROSA. En revanche, toute la partie régulation des variables continues se tient dans la partie physique du holon ressource.

4.2 Définition du Produit

Selon PROSA (Van Brussel et al., 1998), le holon Produit contient des connaissances du processus et du produit pour assurer la réalisation correcte du produit avec une qualité suffisante. La spécification du produit utilisé pour cet article est une spécification orientée services, comme proposé dans (Quintanilla et al., 2014), où le produit est défini par tous les services nécessaires pour l'obtenir (Figure 8). La distinction faite dans cet article par rapport à la définition utilisée dans (Quintanilla et al.,

2014), est que les paramètres et les variables du service peuvent être continus ou discrets.

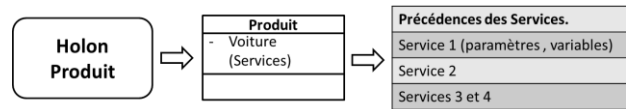


Figure 8. Modèle du produit orienté services.

4.3 Définition de la Ressource

L'holon ressource est composé par une partie physique et une partie logique. La partie logique est une abstraction de la partie physique qui a tous les modèles de conversion d'état discret vers l'état continu et vice-versa (Figure 9). Plusieurs modèles peuvent coexister dans le holon Ressource et représentent les fonctionnalités comme le modèle de détection de défaillance, le modèle de comportement souhaité et tous les modèles qui sont inhérents au paradigme holonique comme le modèle de négociation par exemple.

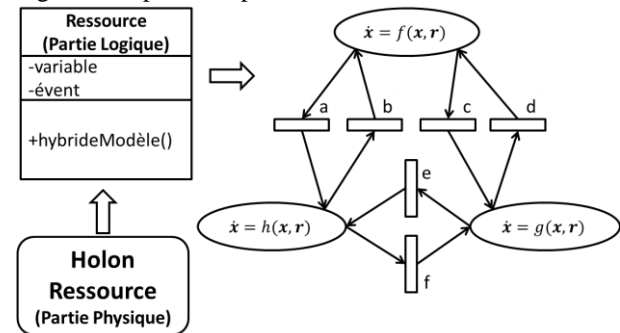


Figure 9. Modèle du holon Ressource.

4.4 Définition du holon Ordre

La fonction et la structure du holon Ordre restent les mêmes que celles proposées par PROSA. En effet, l'architecture proposée dans cet article garantit que toutes les informations et les modèles à être manipulés par le holon Ordre sont discrets ou discrétisés. Le processus de discrétisation des variables de contrôle est classiquement utilisé dans le contrôle des systèmes hybrides. La différence ici est que les modèles de comportement de chaque holon est plus simple à modéliser le comportement global du système hybride, et donc plus simple à discrétiser.

Les systèmes holoniques ont de meilleures performances lorsque le système hybride a un comportement se rapprochant le plus d'un comportement discret. Lorsque ce comportement se rapproche plus d'un système purement continu, il est possible de faire la planification initiale des ressources, mais il est peu probable de pouvoir reconfigurer le système après avoir commencé sans arrêter le système.

5 L'APPLICATION SUR LA CASE D'ETUDE

Le système étudié est le système de commutation précédemment présenté avec $n=3$.

Ainsi, le système a 4 éléments : un serveur et 3 réservoirs. Chaque élément est représenté par un holon Ressource, et un holon Ordre et un holon Produit sont définis pour le contrôle, comme l'illustre la figure 10.

Chaque holon Ressource a deux parties, la partie physique, qui pour des raisons pratiques va être représenté par un programme en Java qui permet la simulation du processus, et une partie logique, qui a le modèle qui permet la transformation de variables continues en variables discrètes. Dans cet exemple, il envoie des alertes au franchissement de seuils (vide ou plein).

Dans ce cas d'étude, le produit est défini par les valeurs des seuils admissibles de chaque réservoirs, représenté par deux vecteurs de nombre réels contenant respectivement les seuils minimum et maximum de chaque réservoir. Enfin, le holon Ordre a la charge de déterminer la position du serveur afin de satisfaire au produit, ce qui va être détaillé dans la section suivante.

5.1 Construction de l'ordonnancement en ligne

Le comportement du cas d'étude a été analysé dans la section 3.2 et il a été constaté qu'il est fortement lié à ses conditions initiales. Les conditions initiales de ce cas d'étude sont les niveaux $x = (0, 0.15, 0.15)^t$ et les sorties des réservoirs $\rho = (0.1, 0.1, 0.1)^t$. Avec ces

conditions initiales, deux réservoirs seront vides à la fois après un temps $\tau = x_i(t)/\rho_i$, (Figure 12). Ceci représente un problème pour le système de contrôle classique, car lorsqu'il choisira l'un des deux réservoirs à remplir, l'autre restera vide a fortiori pendant un certain temps, ce qui n'est pas acceptable pour le système. Si l'analogie du château d'eau présentée section 3.1 est utilisée, alors ceci impliquerait que le quartier serait sans eau.

La solution proposée dans ce document est basée sur la planification de la réservation du serveur, afin d'éviter que deux réservoirs soient vidés en même temps.

En regardant le problème du point de vue de la réservation du serveur avec les conditions initiales présentées à la figure 11, la position initiale du serveur sera sur le réservoir 1 parce qu'il est vide, et lorsque une estimation de temps est fait pour savoir quand les autres réservoirs devront faire la réservation, alors les réservoirs 2 et 3 auront besoin du serveur simultanément, comme montré dans la figure 13. Ceci nous conduit à faire une planification de la réservation. Cette planification pourrait se faire hors-ligne, avec des techniques classiques d'ordonnancement. Toutefois, notre objectif ici est de généraliser ce cas simple à des cas où une grande partie des données sont incertaines, tel l'exemple de reconfiguration présenté dans la section 6.

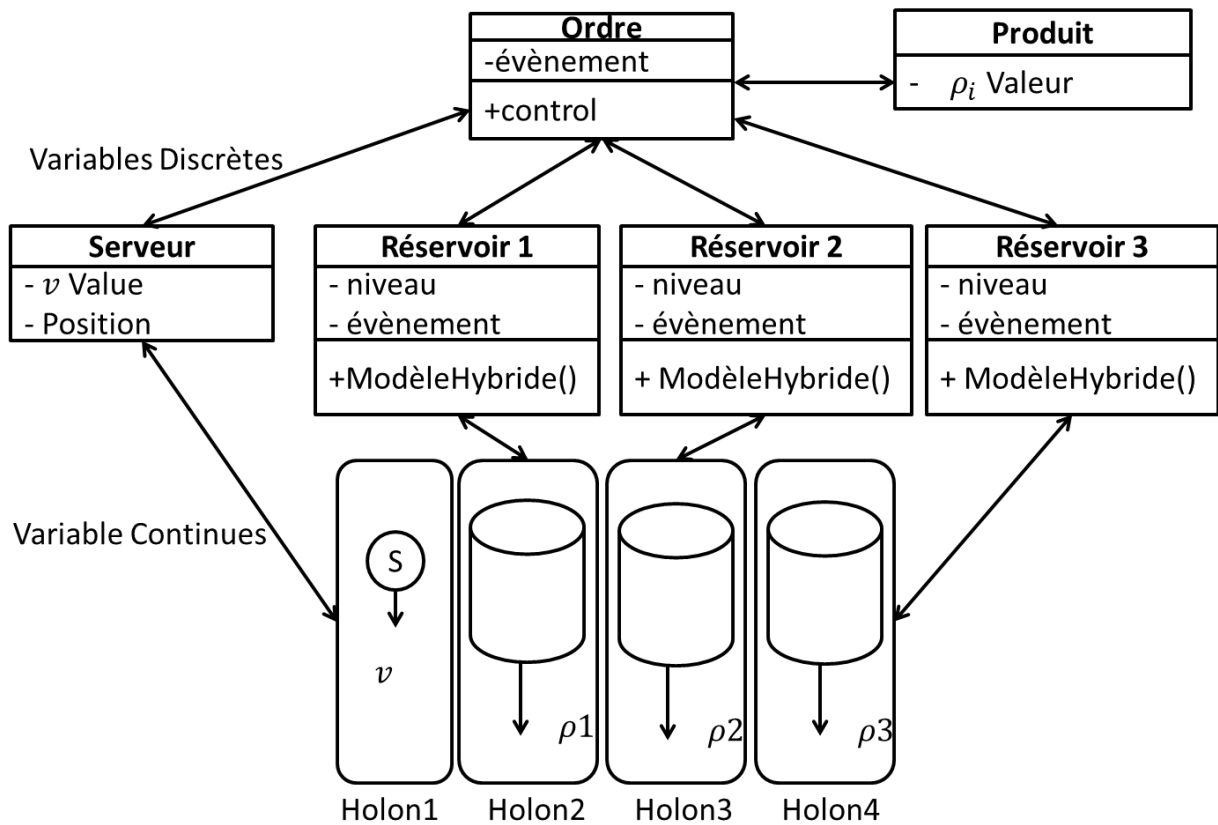


Figure 10. Application du système holonique sur le cas d'étude.

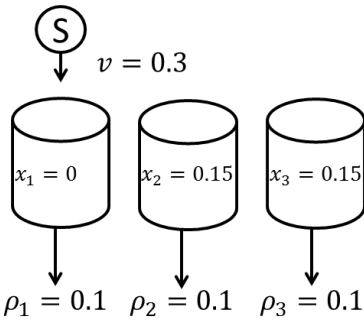


Figure 11. Conditions initiales du système

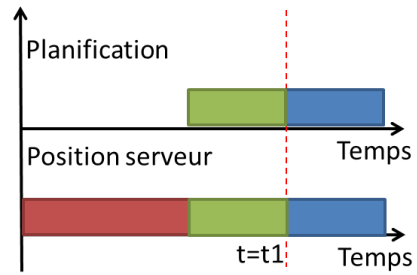


Figure 14. Planification et positionnement du serveur

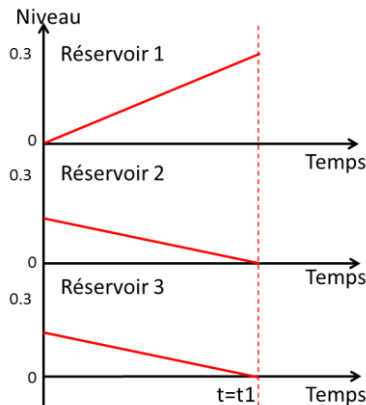


Figure 12. Comportement du niveau des réservoirs.

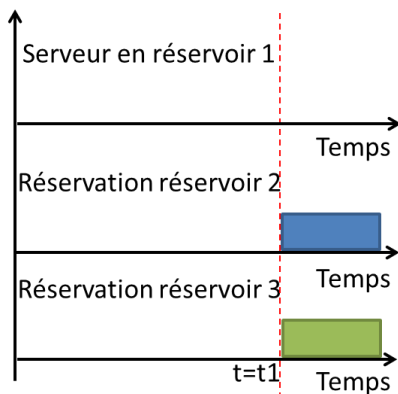


Figure 13. Réservation des réservoirs

Chaque réservoir pose une réservation, à l'exception de celui qui est servi actuellement. En fonction des prévisions de dépassement de seuil inférieur, chacun des autres réservoirs pose une réservation du serveur. Le processus de planification de la réservation consiste à déplacer les réservations qui se chevauchent dans le temps afin d'anticiper au maximum le début du remplissage (Figure 14). Lorsqu'une réservation est atteinte dans le planning, la position du serveur bascule, et le réservoir ayant laissé le serveur planifie le retour du serveur en fonction de son vidage.

6 RESULTATS

Trois configurations de fonctionnement de ce cas d'étude ont été prises en compte :

- **Configuration 1** : trois réservoirs avec les conditions initiales $x = (0, 0.15, 0.15)^t$ et $\rho = (0.1, 0.1, 0.1)^t$ impliquant le vidage simultané de deux réservoirs très rapidement;
- **Configuration 2** : dix réservoirs et les conditions initiales suivantes

$$x = \begin{pmatrix} 0, 0.2, 0.1, 0.1, 0.1, 0.1 \\ 0.1, 0.1, 0.1, 0.1, 0.1 \end{pmatrix}^t$$
 et

$$\rho = \begin{pmatrix} 0.1, 0.1, 0.1, 0.1, 0.1 \\ 0.1, 0.1, 0.1, 0.1, 0.1 \end{pmatrix}^t$$
, où huit réservoirs seraient simultanément vides ;
- **Configuration 3** : un système qui commute de la première vers la seconde configuration au bout de 600 secondes de fonctionnement.

6.1 Courbes résultats

La figure 15 montre le fonctionnement du système dans la configuration 1 sur un horizon de 1000 secondes. Le cercle positionné à 180 secondes indique le moment où le deuxième réservoir commence à se remplir en anticipant sa réservation, et laisser la réservation au plus tard du troisième réservoir.

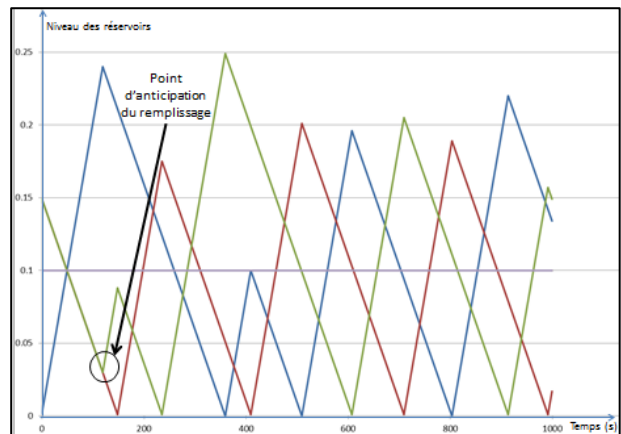


Figure 15. Configuration n°1.

Une analyse similaire peut être faite sur la configuration n°2 (Figure 16). Il est notable que beaucoup de réservations sont anticipées, mais que chaque réservation est plus courte. En effet, la conséquence de l'hypothèse du système fermé est que le débit du serveur augmente avec le nombre de réservoirs à remplir ce qui diminue le temps nécessaire à remplir un réservoir, et que diminuer la durée de réservation rend plus flexible la mise en place du planning.

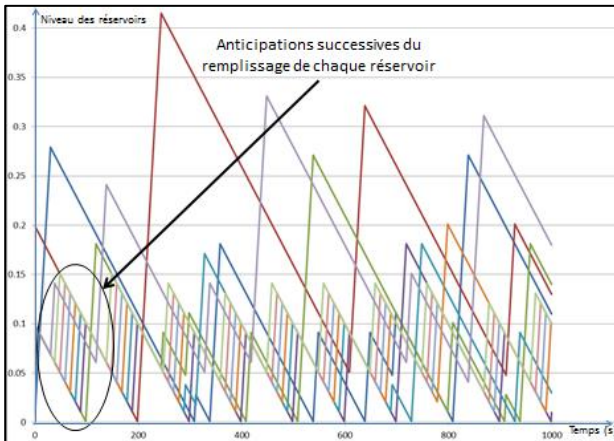


Figure 16. Configuration n°2.

La configuration n°3 a pour objectif de démontrer la flexibilité du système. La figure 17 met en évidence la capacité du système de contrôle à se reconfigurer. A $t=600$, lorsque les nouveaux réservoirs sont introduits, un nouveau planning est calculé et le système continue à fonctionner en temps réel. La Figure 18 indique la position du serveur. A de très rares exceptions près, on observe que la position du serveur est relativement stable.

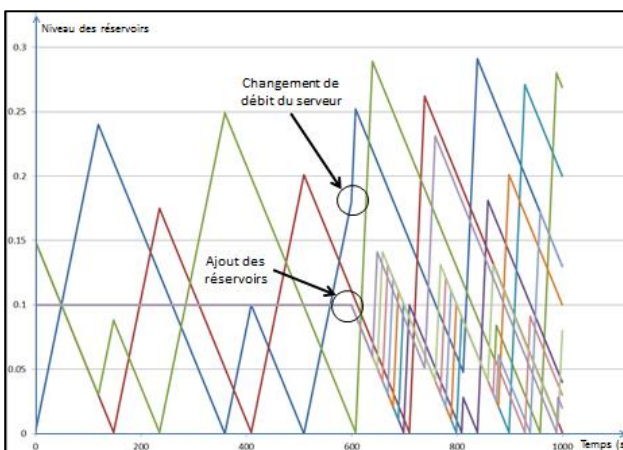


Figure 17. Configuration n°3.

6.2 Discussion

Les résultats obtenus dans la section précédente montrent qu'un système hybride peut être contrôlé par un contrôle basé sur le paradigme holonique. Les restrictions d'états initiaux et le problème de non-déterminisme (deux réservoirs vides à la fois) ont été également supprimés avec

l'aide de la planification en ligne. Ce mode de contrôle permet d'implanter un système sous-optimal mais respectueux des critères de qualité exprimés dans le holon Produit avec des modèles de comportement simples.

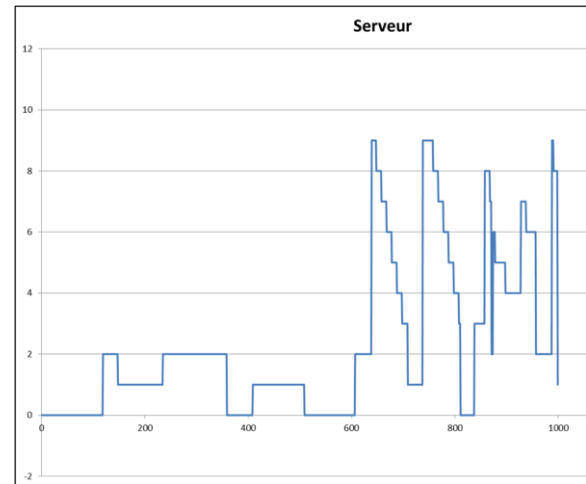


Figure 18. Positionnement du serveur.

7 CONCLUSION

Le travail proposé dans cet article concerne l'implémentation d'une architecture holonique sur les systèmes hybrides. Les résultats présentés tendent à valider la bonne adéquation de cette architecture sur cette classe de systèmes. L'architecture holonique fournit également ses intrinsèques simplicité, avec la gestion de modèles de comportement plus simples, et flexibilité, avec la possibilité d'ajouter de nouvelles ressources en ligne sans arrêter le système.

Les résultats sont limités à l'exécution d'un processus et ne s'étendent pas jusqu'à la planification préalable des ressources. Par conséquent, la poursuite de ces travaux permettra de couvrir les aspects de planification des ressources et la tolérance aux pannes, ainsi que l'adaptation d'autres architectures de référence holoniques, telles qu'ADACOR ou HCBA.

REFERENCES

- Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L., and Peeters, P. (1998). Reference architecture for holonic manufacturing systems: PROSA. *Comput. Ind.* 37, 255–274.
- Chacón, E., Besembel, I., Narciso, F., Montilva, J., and Colina, E. (2002). An integration architecture for the automation of a continuous production complex. *ISA Trans.* 41, 95–113.
- Chase, C., Serrano, J., and 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.
- Chirn, J.-L., and McFarlane, D.C. (2000). A holonic component-based approach to reconfigurable manu-

- facturing control architecture. In 11th International Workshop on Database and Expert Systems Applications, 2000. Proceedings, pp. 219–223.
- Chokshi, N., and McFarlane, D. (2007). A Distributed Coordination Approach to Reconfigurable Process Control (Springer).
- Chokshi, N., and McFarlane, D. (2008). A distributed architecture for reconfigurable control of continuous process operations. *J. Intell. Manuf.* 19, 215–232.
- Koutsoukos, X.D., Antsaklis, P.J., Stiver, J.A., and Lemmon, M.D. (2000). Supervisory control of hybrid systems. *Proc. IEEE* 88, 1026–1049.
- Labinaz, G., Rudie, K., Ricker, L., Sarkar, N., and Bayoumi, M.M. (1996). A hybrid system investigation of fluid-filled tanks. *Math. Theory Netw. Syst.* 96–01.
- Labinaz, G., Bayoumi, M.M., and Rudie, K. (1997). A survey of modeling and control of hybrid systems. *Annu. Rev. Control* 21, 79–92.
- Leitão, P., and Restivo, F. (2006). ADACOR: A holonic architecture for agile and adaptive manufacturing control. *Comput. Ind.* 57, 121–130.
- McFarlane, D.C. (1995). Holonic manufacturing systems in continuous processing: concepts and control requirements. In *Proceedings of ASI*, (Citeseer), p. 282–273.
- Quintanilla, F.G., Cardin, O., and 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*, T. Borangiu, D. Trentesaux, and A. Thomas, eds. (Springer International Publishing), pp. 177–193.
- Tian, Y.-P. (2005). Delayed feedback control of chaos in a switched arrival system. *Phys. Lett. A* 339, 446–454.
- Ushio, T., Ueda, H., and Hirai, K. (1995). Controlling chaos in a switched arrival system. *Syst. Control Lett.* 26, 335–339.