THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Institut National des Sciences Appliquées de Toulouse (INSA Toulouse)

Discipline ou spécialité :

Informatique - Intelligence Artificielle

Présentée et soutenue par :
Sylvain Videau

le : mercredi 6 juillet 2011

Titre :

Contrôle de processus dynamiques par systèmes multi-agents adaptatifs : application au contrôle de bioprocédés

---

JURY

Salima Hassas Professeur, Université Claude Bernard-Lyon 1 Rapporteur
Ivan Marc Directeur de Recherche au CNRS, LRGP Rapporteur
Marie Beurton-Aimar Maître de conférences, Université Bordeaux 1 Examinateur
Marie-Pierre Gleizes Professeur, Université Toulouse 3 Examinateur
Jean-Louis Uribelarrea Professeur, INSA Toulouse Directeur de thèse
Carole Bernon Maître de conférences, Université Toulouse 3 Co-directrice de thèse
Pierre Glize Ingénieur CNRS HdR, IRIT Invité

Ecole doctorale :
Mathématiques Informatique Télécommunications (MITT)

Unité de recherche :
IRIT / LISBP

Directeur(s) de Thèse :
Jean-Louis Uribelarrea Professeur, INSA Toulouse Directeur de thèse
Carole Bernon Maître de conférences, Université Toulouse 3 Co-directrice de thèse

Rapporteurs :
Salima Hassas Professeur, Université Claude Bernard-Lyon 1 Rapporteur
Ivan Marc Directeur de Recherche au CNRS, LRGP Rapporteur

Remerciements

à Carole Bernon, ma directrice de thèse, pour ta patience extrême et toute l'aide apportée au cours de ces dernières années.

à Jean-Louis Uribelarrea, mon directeur de thèse, pour ton ouverture d'esprit sur les approches informatiques, la confiance que tu m'as accordé et la liberté de recherche dont j'ai bénéficié.

à Pierre Glize, car ce travail te doit énormément. Pour ton soutien, tes idées et leur faculté à cultiver l'envie d'explorer toujours un peu plus.

à l'équipe SMAC (et aux doctorants en particulier) pour cette ambiance d'échange, motivante et l'enthousiasme communicatif.

à ma mère, pour tant de choses qu'il serait insultant d'en faire la liste.

à mes amis, Greg, Marc, Benoit, Flow, Claire et Laurent, pour Tout.

et à ces mots qui ont touchés juste.

A

Table des matières

Chapitre I

Introduction générale

Parler du contrôle de systèmes est une tâche complexe à plusieurs titres. De par la généricité du terme contrôle tout d'abord, propice à une mauvaise interprétation du problème considéré dans cette thèse, et de par l'étendue des travaux réalisés à ce sujet, provenant de nombreux domaines de recherche, au sein desquels il est difficile de distinguer un approche commune. Cette pluridisciplinarité apparaît clairement dans le problème dont il est question ici : proposer un système informatique générique capable de contrôler des procédés biologiques.

Actuellement, la conception de tels systèmes nécessite un investissement important de l'expert du domaine d'application, en particulier dans l'ajustement et la configuration des méthodes de contrôle employées ainsi que dans la modélisation des informations nécessaires au fonctionnement du système de contrôle. Dans certains cas, cette tâche est tout simplement trop complexe et ne peut s'opérer que sous la supervision d'un opérateur humain.

Ce phénomène provient essentiellement du manque de généricité des approches usuelles nécessitant une importante étape d'instanciation avant de pouvoir être en mesure de contrôler un procédé précis, ainsi que de leur manque d'adaptation rendant ces systèmes de contrôle particulièrement sensibles aux différents bruits.

L'objectif de cette thèse est donc à la frontière de l'informatique et de la biologie, à savoir la conception d'un système de contrôle suffisamment générique et adaptatif pour réduire de manière très significative les connaissances biologiques nécessaires au contrôle.

Ainsi, les contributions des travaux présentés dans ce manuscrit s'étendent des apports informatiques théoriques quant à la conception de tels systèmes de contrôle, jusqu'aux apports pratiques concernant les avantages perçus par les utilisateurs finaux.

I.1 Le choix de l'étude des bioprocédés

La complexité du contrôle de bioprocédés en fait donc un sujet incontournable pour observer les limites des approches de contrôle usuelles. Sa dynamique, tout comme le manque de connaissance sur le fonctionnement même du système à contrôler, sont des caractéristiques rejoignant des problématiques centrales en informatique, et en intelligence artificielle en particulier, telles que l'apprentissage et l'auto-adaptation des systèmes.

De plus, dans l'optique du développement d'un système de contrôle générique, l'importante variété de systèmes différents réunis sous le nom de bioprocédés est un atout permettant de disposer d'un nombre conséquent d'exemple d'applications, et donc, de dynamiques à étudier.

L'aspect pluridisciplinaire de ce travail doit également permettre au système développé d'apporter à l'utilisateur, au-delà des solutions de contrôle, des informations pertinentes sur les actions qu'il réalise ainsi que sur les raisons de ces choix.

Ces points contribuent à positionner le contrôle de bioprocédés comme l'application la plus pertinente afin de tester les techniques de contrôle informatique développées dans cette thèse, mais également de guider leur développement sous un ensemble d'hypothèses imposant une grande généricité tout en restant proche de la pratique du contrôle réel. Il ne s'agit pas d'un travail visant à fournir une solution optimale et biologiquement pertinente pour un seul bioprocédé donné, mais il s'agit d'utiliser la complexité inhérente aux procédés biologiques afin de pouvoir concevoir un système de contrôle générique et d'étudier son comportement dans des conditions extrêmes.

La problématique scientifique découlant de ce domaine d'application peut être exprimée comme la conception d'une architecture informatique capable de s'adapter sans modèle préalable, à la dynamique des bioprocédés tout offrant la robustesse adéquate à la gestion des différents bruits inhérents à la manipulation de tels systèmes biologiques.

I.2 Organisation du manuscrit

Outre cette introduction, ce document est organisé en cinq chapitres qui ont pour vocation de guider le lecteur dans la démarche entreprise : de la présentation générale du contexte de cette étude au bilan relatif aux approches proposées pour la conception du système de contrôle générique cible.

Tout d'abord, la définition du contexte de l'étude permet de positionner le travail réalisé vis-à-vis des différents domaines concernés. Ainsi, l'ensemble des définitions liées au contrôle et des technologies employées est détaillé, fournissant les bases sur lesquelles notre problématique s'articule accompagnées des informations nécessaires à la compréhension du choix de notre approche de résolution.

Dans un second temps, l'état de l'art des méthodes de contrôle est dressé, permettant ainsi de caractériser précisément les limites et les verrous rencontrés dans les approches usuelles. Un large panorama de ces approches est présenté à partir de l'étude des méthodes de contrôle provenant de domaines d'origines variées.

La partie suivante décrit l'approche par Système Multi-Agents Adaptatifs tout en justifiant les raisons de ce choix technologique et ses conséquences sur le développement informatique permettant de résoudre le problème du contrôle. Cette description nécessaire permet également de positionner précisément la problématique du contrôle d'un point de vue informatique, ainsi que les apports de cette thèse quant aux verrous théoriques existants. Les hypothèses formant la base du développement logiciel effectué sont donc présentées.

Une fois ces fondations théoriques posées, les deux chapitres suivants détaillent la conception et l'application de deux systèmes créés afin d'atteindre cet objectif de contrôle générique. Chacun d'entre eux se base sur des ensembles d'hypothèses différents, qui ont été détaillés dans la partie précédente, et sont évalués sur la résolution d'un ensemble de problèmes permettant de vérifier leurs caractéristiques ainsi que leur adéquation vis-à-vis de la problématique générale du contrôle de bioprocédés.

La dernière partie de cette thèse est, pour sa part, consacrée à la synthèse des résultats obtenus, ainsi qu'au détail des apports effectifs du travail réalisé. Les perspectives d'évolution et d'extension des approches présentées sont aussi discutées.

Chapitre II

Contexte de l'étude

Comme nous l'avons vu dans le chapitre introductif de cette thèse, le problème qui se pose à nous se situe au carrefour de deux domaines : le contrôle de systèmes complexes, en particulier le contrôle de bioprocédés, mais aussi l'informatique puisque les solutions proposées et mises en œuvre s'appuieront sur la technologie des systèmes multi-agents. C'est la raison pour laquelle l'objet de ce second chapitre est de présenter le contexte de cette étude en définissant plus précisément les notions liées à ces deux domaines.

Dans un premier temps, ce chapitre dresse ainsi un historique de la problématique du contrôle de manière à fournir les éléments essentiels dans la compréhension de l'aspect pluridisciplinaire du contrôle et d'introduire les concepts-clés qui seront définis par la suite. Ces concepts se divisent en deux catégories, biologique et mathématique, et couvrent les notions nécessaires pour permettre la catégorisation des approches présentées dans le chapitre suivant qui détaillera l'essentiel des techniques employées afin de contrôler différents procédés complexes.

La seconde partie de ce chapitre est consacrée à la définition des systèmes multi-agents qui représentent l'une des technologies possibles pour mettre en place un système de contrôle de procédés complexes et dynamiques. Les caractéristiques de base des agents et des systèmes à base d'agents sont ainsi fournies afin d'illustrer, là encore, certaines des approches présentées dans le chapitre consacré à l'état de l'art proprement dit.

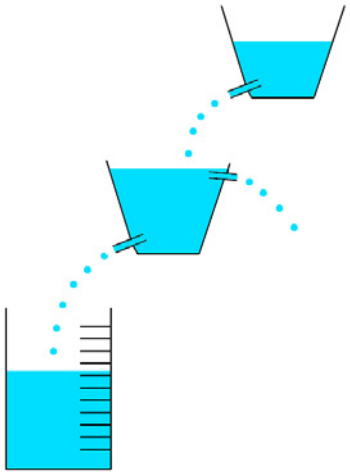

Figure II.1 – Principe de la clepsydre grecque de Ctesibios

II.1 Contrôle de systèmes complexes

II.1.1 Historique du contrôle

Depuis des milliers d'années, l'homme cherche à contrôler son environnement, et recherche donc les moyens d'agir sur celui-ci en vue d'améliorer sa condition actuelle.

A l'origine, la problématique du contrôle de systèmes relevait de l'ingénierie pure, et ses applications se trouvaient fortement liées aux problèmes pratiques auxquels l'humanité était alors confrontée.

L'un des premiers de ces problèmes fut la mesure précise du temps, et bien que de nombreuses méthodes furent développées pour parvenir à cette fin, en particulier la clepsydre égyptienne, ce furent les grecs qui y appliquèrent un raisonnement de contrôle, afin de surmonter ses défauts et d'améliorer sa précision. En effet, Ctesibios proposa vers 300 av. J-C. une solution au problème du débit de la clepsydre illustrée par la figure II.1. Ce problème naquit de la constatation suivante : selon la quantité d'eau dans le bol supérieur, le débit de sortie variait du fait de la différence de pression exercée par le volume d'eau. Ainsi, durant son utilisation, le débit variable réduisait la précision des mesures. Ctesibios modifia alors le fonctionnement de la clepsydre en rajoutant un bol intermédiaire, dont le niveau est maintenu constant à l'aide d'un mécanisme de rejet du surplus d'eau. Ainsi, le bol servant à la mesure proprement dite est alimenté de manière constante.

Cet exemple simple souligne les efforts fournis afin de contrôler un système, c'est-à-dire,

alimenter celui-ci de manière à obtenir le résultat attendu.

De nombreuses années plus tard, la révolution industrielle imposa l'utilisation de mécanismes de contrôle automatique des machines, ceux-ci ne pouvant pas forcément être réalisés à la main. Ainsi, les régulateurs de température, pression ou niveau furent développés et améliorés afin de permettre l'utilisation massive de machines. Cependant, ces régulateurs étaient principalement issus d'essais-erreurs, et portés par l'intuition de leurs concepteurs.

Il fallut attendre les années 1800 afin de trouver l'origine de la théorie mathématique du contrôle. Celle-ci fut soutenue par l'utilisation d'équations différentielles dans l'analyse de l'instabilité observée dans un système de contrôle de télescope par G.B. Airy en 1840. Ce furent cependant les travaux de James Clerk Maxwell en 1868, sur les régulateurs à boules, qui conduisirent aux prémices de ce qui deviendra la théorie du contrôle.

Le début des années 1900 apporta pour sa part l'apparition de la théorie des systèmes. Jusqu'alors, la notion même de système était absente du contrôle, les notions d'entrées et sorties, reliées entre elles par une entité dynamique et plongée dans un environnement, firent leur apparition et devinrent des concepts-clés.

Les années 1900 marquèrent également plusieurs changements radicaux liés, d'une part, à l'essor de la communication de masse et, d'autre part, aux guerres mondiales.

L'essor des communications et les problématiques d'amplification du signal qui y sont liées entraînèrent la découverte de la notion de rétroaction négative par H.S. Black, et l'approche par domaine fréquentiel, soutenue par P.-S. de Laplace, J. Fourier et A.L. Cauchy.

De leur côté, les guerres mondiales introduisirent les problématiques de navigation d'engins et de gestion de visée de missiles. Il est intéressant de noter que les techniques développées alors, telles que les régulateurs PID présentés dans la partie III.1, sont toujours d'actualité.

La fin de la seconde guerre mondiale coïncida avec la maturité du champ du contrôle des systèmes, l'ensemble des ouvrages écrits jusqu'alors constituant ce que l'on nomme la théorie classique du contrôle.

Cependant, la théorie classique, bien que suffisante pour nombre de problèmes, rencontra des difficultés à contrôler des systèmes non-linéaires et possédant de multiples entrées et sorties.

Ainsi, un retour sur les équations différentielles fut opéré et le domaine du contrôle dit « optimal » abondamment étudié, en particulier grâce aux travaux de R. Kalman. Cette période coïncida avec l'essor de l'informatique et ses capacités à résoudre des équations complexes. Cette approche, centrée sur la mise en équation du contrôle et l'utilisation d'un modèle du système à

contrôler, est nommée le contrôle « moderne ». Les décennies 70 et 80 rapprochèrent alors les approches classiques et modernes du contrôle, afin de concilier le meilleur des deux mondes et de profiter des points forts de chaque approche.

Ce bref historique permet de souligner l'évolution des besoins de contrôle, entraînant l'application de cette notion à des systèmes de plus en plus complexes, dynamiques et, de ce fait, la création de méthodes de contrôle différentes, par les spécialistes des domaines concernés.

II.1.2 Définitions liées au contrôle

Avant de présenter les techniques de contrôle communément employées, il est nécessaire de définir le contrôle lui-même, ainsi que le contexte dans lequel le terme est utilisé. En effet, parler de contrôle couvre un large échantillon d'actions spécifiques détaillées dans les paragraphes suivants.

II.1.2.1 Notions de base

Contrôler un système revient à déterminer les modifications à effectuer sur certaines de ses entrées, en vue d'obtenir un résultat donné. À partir de cette définition sommaire émerge une grande variété de sous-problèmes qui sont liés aux différentes contraintes apparaissant dans les modifications possibles et à la représentation éventuelle des connaissances possédées sur le système à contrôler. De même, la généricité du problème de contrôle entraîne son apparition dans de multiples contextes, dérivant des problématiques de recherches diverses et impliquant une forte pluridisciplinarité.

Le contrôle de systèmes n'est pas un problème nouveau en soi, mais l'évolution de l'informatique a entraîné une plus grande facilité dans la manipulation de systèmes dits « complexes ». En effet, l'informatique est un outil très puissant pour la simulation de systèmes, quelle que soit leur nature : biologiques, économiques, sociaux, etc. La complexité de tels systèmes provient de l'impossibilité à prévoir leur comportement ou leurs évolutions, ainsi que du nombre d'entités en interaction, deux points se prêtant particulièrement à la modélisation informatique. Le problème du contrôle de systèmes complexes se développa donc de pair avec cette phase de modélisation devenue plus accessible et plus puissante.

Le contrôle de systèmes complexes est donc un domaine abondamment étudié, car au carrefour de nombreux champs de recherche tels que l'optimisation, l'automatique, la résolution

de problèmes inverses ou la théorie mathématique du contrôle. De ce fait, selon le champ d'application considéré, nous obtenons une vue bien particulière du contrôle, liée d'une part aux spécificités du problème sur lequel il est appliqué, mais également à la manière d'aborder ledit contrôle.

Si nous considérons par exemple le contrôle d'un procédé biologique ou d'une chaîne de production, les mécanismes habituellement mis en œuvre, bien que servant un objectif similaire, présentent des différences importantes. Ces distinctions proviennent des contraintes liées aux problèmes mais, également, d'une évolution distincte des techniques de contrôle, héritées d'une base mathématique commune mais ayant au fil du temps connu des spécialisations à des domaines précis. L'historique, réalisé dans la partie précédente, témoigne de ce fait, et évoque des approches fondamentalement différentes selon les problématiques cibles du contrôle.

Ainsi, de nombreux termes couramment utilisés lorsque l'on considère le contrôle présentent des définitions sensiblement différentes selon les domaines d'applications. De plus, la définition générique du contrôle décrite au début de ce paragraphe entraîne une apparition de plusieurs « contrôles »à différents niveaux d'un même problème. C'est-à-dire que, selon le point de vue d'un observateur extérieur au système, un problème de « contrôle » peut faire intervenir plusieurs sous-problèmes faisant, eux aussi, appel à des notions de « contrôle » à une échelle plus réduite, répondant à des définitions précises issues du domaine de l'Automatique.

Prenons l'exemple d'un bioprocédé (considéré ici comme un procédé quelconque reposant sur l'utilisation de réactions biologiques) produisant une certaine quantité d'un élément donné. L'utilisateur désire augmenter cette quantité produite. Pour ce faire, différentes actions doivent être entreprises durant l'exécution du bioprocédé, afin de le conduire vers le résultat souhaité. Ces actions constituent ici des « commandes », effectuées afin d'atteindre les « consignes » établies par la « conduite » du procédé, l'application correcte de ces « commandes » étant assurée par des mécanismes de « contrôle », le tout étant soumis à une « supervision » humaine ou informatique.

Cette phrase souligne clairement la distinction entre le contrôle au sens considéré dans ce travail, et le contrôle spécifique au domaine des bioprocédés : nous parlons ici du contrôle comme de l'ensemble composé des mécanismes de contrôle, conduite, commande et supervision au sens Automatique de ces termes. En ce sens, nous nous rapprochons du terme « control » anglais qui, comme le détaillera l'état de l'art, regroupe ces différentes significations.

Dans cette thèse, nous nous intéresserons donc au contrôle dans son ensemble, c'est-à-dire, des techniques locales et réactives permettant de compenser des modifications de certaines

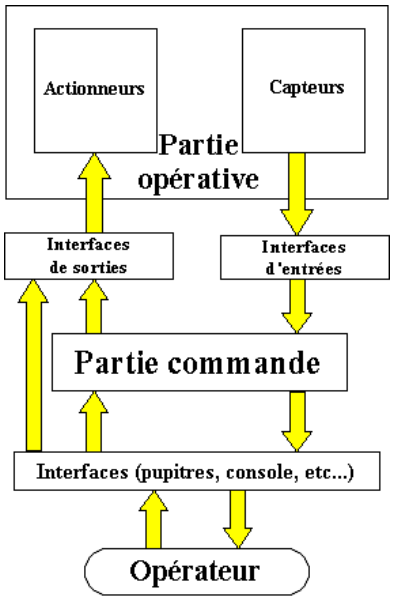

Figure II.2 – Principe général du contrôle

valeurs jusqu'aux méthodes impliquant l'utilisation de modèles complexes en vue de décider des actions à entreprendre pour atteindre un objectif calculé ou fourni par l'utilisateur. Ceci regroupe donc les techniques employées dans la partie de commande du procédé, ainsi que certains mécanismes de fonctionnement appartenant à la partie opérative, tel que l'illustre la figure II.2. De ce fait, les méthodes qui seront présentées dans l'état de l'art du chapitre suivant proviendront indifféremment du contrôle moderne ou classique afin de donner une vision générale des différentes approches adoptées.

II.1.2.2 Le contrôle du point de vue de la biologie

Une fois la problématique du contrôle posée, il est nécessaire de s'intéresser à son contexte d'application, à savoir les bioprocédés. Un bioprocédé est un procédé quelconque basé sur des réactions biologiques afin d'obtenir une production définie, de telles réactions pouvant, par exemple, utiliser des micro-organismes ou encore des enzymes. Ainsi, un bioprocédé peut aussi bien être une production industrielle de yaourts que le traitement d'eaux usées, en passant par la production chimique pharmaceutique. Cependant, malgré une telle diversité dans la taille et le mode de fonctionnement des bioprocédés, le fonctionnement général de ceux-ci reste identique. Cette production est rendue possible par l'ajout de ressources consommées par les micro-organismes ou déclenchant la réaction biologique attendue. Cet ajout peut être réalisé au fur et à mesure de la réaction, et donc être limitant pour celle-ci dans un procédé dit « fed-

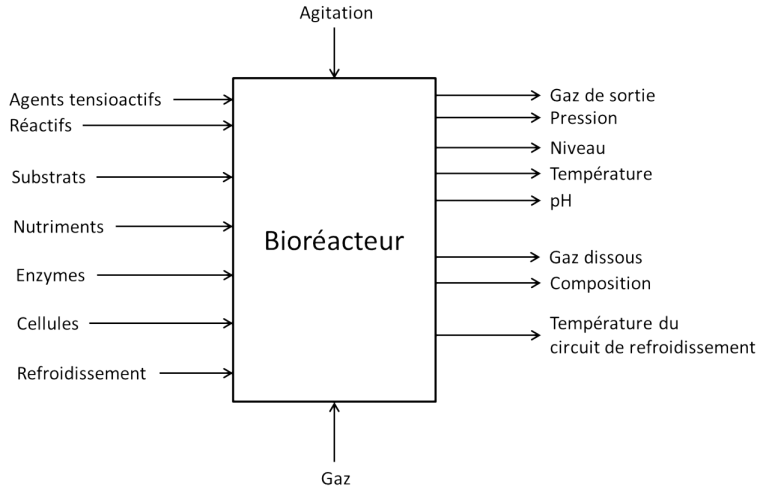

Figure II.3 – Entrées et sorties d'un bioréacteur

batch ». L'ajout de ressources peut également être dit « continu » et non limitant. Enfin, dans le cas d'un procédé dit « batch », la quantité de ressources est définie au début du procédé et n'est pas modifiée par la suite.

Le contrôle, pour sa part, fait intervenir les notions d'entrées et de sorties. Celles-ci désignent les variables sur lesquelles l'utilisateur peut agir et les variables simplement mesurées. Il ne s'agit pas d'entrées/sorties au sens séquentiel, impliquant un ordre précis de modifications menant de ces premières aux dernières. La figure II.3 illustre ce fait. Les entrées et sorties sont donc simplement des quantités mesurables à un instant donné par un ensemble de capteurs placés sur le bioprocédé, la différence entre ces deux notions provenant du fait que l'on peut agir directement sur la valeur des entrées.

Ces variables de sorties sont également divisées en deux types, dits « en-ligne » et « hors-ligne ». Une variable est dite « en-ligne » si sa valeur est mesurée directement sur le bioprocédé à l'aide de capteurs. À l'inverse, une variable « hors-ligne » est obtenue à l'aide d'un calcul effectué à partir d'autres valeurs. Ce type de calcul permet, dans certains cas, de surmonter l'absence de capteur dédié au suivi d'une variable clé, au prix d'un ajout de bruit supplémentaire.

Contrôler un bioprocédé revient à maintenir un environnement quasi-optimal pour permettre la croissance des micro-organismes attendus, tout en limitant et supprimant l'apparition de produits toxiques pour ceux-ci.

Cependant, la complexité de cette tâche est liée à la complexité du bioprocédé lui-même, dérivant de la quantité d'éléments et d'interactions entrant en ligne de compte, ainsi que de la dimension temporelle non négligeable qui ajoute de l'incertitude quant aux résultats mesurés.

En effet, la présence de délais entre le moment où une action de contrôle est effectuée et celui où son résultat est mesuré entraîne une quantité de bruit importante qu'il est nécessaire de savoir distinguer et gérer. Ces délais sont appelés des « dead-time » et peuvent varier pendant l'exécution du bioprocédé lui-même, ce qui complexifie encore la prévision des états futurs de celui-ci. La difficulté liée à la création d'un modèle d'un tel bioprocédé provient de ces phénomènes.

À cela s'ajoute le faible nombre de mesures « en-ligne » disponibles ce qui limite les indicateurs visibles des conséquences des contrôles effectués et impose à l'observateur de s'appuyer sur des données inférées afin de pouvoir décrire l'état biologique du système. Cette limite dérive de l'absence de certains capteurs dans les installations accueillant les bioprocédés ou, plus généralement, de l'impossibilité technique de mesurer la donnée souhaitée. Malgré les énormes progrès réalisés dans ce domaine au cours des trente dernières années, certaines données restent inaccessibles car il est impossible de les mesurer en temps réel et de manière non-intrusive, c'est-à-dire sans modifier le bioprocédé en cours.

Nous remarquons donc que le contrôle de bioprocédés se heurte à de multiples limites, provenant aussi bien du niveau matériel que théorique. Ces limites entraînent en pratique la nécessité d'instancier au bioprocédé cible les techniques de contrôle appliquées, aucune méthode de contrôle générique n'étant suffisante pour effectuer cette étape elle-même.

Globalement, le contrôle de bioprocédés couvre donc l'ensemble des problématiques abordées en Automatique, tout en imposant de lourdes contraintes de par sa complexité et son nombre limité de données utilisables.

II.1.2.3 Le contrôle du point de vue des mathématiques

Lorsque nous parlons de contrôle, il est naturel d'évoquer les théories mathématiques sous-jacentes et de définir les termes les plus utilisés quel que soit le domaine d'application du contrôle. Cette étape nous permet donc de caractériser les techniques qui seront appliquées à la problématique définie dans les paragraphes précédents.

Un système de contrôle forme une boucle qui est dite « ouverte » ou « fermée » selon sa nature.

– Une boucle « ouverte » signifie qu'il n'y a pas de connexion directe entre la sortie de ce système et ses entrées. Le contrôle est effectué sans aucune rétroaction relative à l'état des sorties, et se base seulement sur les connaissances du système à contrôler implicitement

incluses dans le contrôleur par l'expérimentateur.
– Une boucle « fermée » signifie que le contrôleur possède cette rétroaction, cette vue sur les sorties, et qu'il peut ainsi contrôler les entrées en fonction des réactions observées.

Dans les deux cas, il est nécessaire de définir une fonction permettant de déterminer le contrôle à appliquer. Cette fonction est appelée la « loi du contrôle ».

Une autre distinction utile, afin de préciser le problème, est la classification du système à contrôler selon son caractère linéaire ou non. Un système non linéaire peut posséder de multiples points d'équilibre et des variations de solutions au cours du temps. La valeur de ses sorties n'est pas proportionnelle à la valeur de ses entrées. On ne peut pas obtenir sa (ses) valeur(s) de sortie à partir d'une combinaison linéaire de ses composants, ce qui rend généralement ce type de systèmes bien plus complexe que ses équivalents linéaires et implique l'utilisation de techniques spécifiques afin de conduire son étude.

Une fois le système à contrôler caractérisé, il convient de sélectionner le type de système de contrôle à appliquer. Cette étape dépend du type de contrôle souhaité, en fonction du nombre de variables à réguler par exemple, ainsi que des connaissances acquises ou accessibles en cours d'exécution sur le procédé à contrôler lui-même. Les deux caractéristiques principales en question sont l'optimalité et la robustesse.

Un contrôle peut être « optimal » s'il minimise une fonction de coût définie. Ainsi, la conception d'un système de contrôle optimal est centrée sur cette notion de fonction de coût. Un exemple classique de contrôle optimal revient à minimiser la consommation d'essence d'un véhicule devant se rendre d'un point A à un point B.

Enfin, un contrôle est dit « robuste » s'il est conçu afin de gérer l'incertitude. En effet, les systèmes de contrôle robustes doivent être capables de fonctionner tant que l'incertitude concernant certains paramètres reste dans des bornes définies. Cette notion de robustesse n'est donc valide que pour un ensemble de valeurs donné, et assure la stabilité du système dans cette plage-ci. Ces définitions se complètent, permettant à un système de contrôle d'être à la fois optimal et robuste.

Dans le cadre de nos travaux sur les bioprocédés, nous abordons le contrôle robuste de systèmes dynamiques et non linéaires, à l'aide d'une boucle fermée. L'objectif est de contrôler l'évolution de certains éléments produits par un bioprocédé à l'intérieur d'un bioréacteur.

II.2 Les systèmes multi-agents

Comme nous venons de le voir, le contrôle de bioprocédés impose l'utilisation de méthodes de contrôle à la fois robustes et adaptatives. La technologie des Systèmes Multi-Agents (SMA) peut constituer une solution envisageable pour mettre en œuvre un système de contrôle pourvu de ces deux caractéristiques. C'est la raison pour laquelle, la présentation de ces systèmes, issus de l'Intelligence Artificielle (IA), est à présent effectuée.

Un système multi-agent est un système, artificiel ou non, constitué de plusieurs entités autonomes nommées agents. L'origine de tels systèmes provient de l'Intelligence Artificielle Distribuée (IAD), discipline se proposant de répondre aux verrous rencontrés par l'intelligence artificielle tels que la complexité croissante des systèmes réalisés, et de bénéficier des avancées technologiques apparues en particulier dans les possibilités de distribution des calculs. Ainsi, trois champs d'études se sont distingués : le premier d'entre eux est la résolution distribuée de problèmes visant à définir des techniques permettant de diviser la résolution d'un problème donné entre plusieurs entités communicantes. Le second domaine est l'intelligence artificielle parallèle, dont l'objectif est de répondre aux contraintes de performances en proposant des approches de calculs parallèles ainsi que des algorithmes adaptés à cette approche. Enfin, l'approche des systèmes multi-agents propose de résoudre des problèmes par le biais de la coordination d'entités intelligentes nommées agents. Ainsi, les trois champs de l'IAD permettent la création de systèmes dont les connaissances ainsi que les traitements sont distribués et dont le contrôle est décentralisé. Les systèmes monolithiques issus des approches classiques de l'intelligence artificielle sont donc fragmentés et l'utilisation de la notion d'agent apparaît comme base de ces nouveaux systèmes composés d'entités indépendantes et communicantes. Ce phénomène conduit à la disparition de la nécessité de résolution globale d'un problème complexe au profit d'une décomposition de ce dernier en sous-problèmes dont la résolution est déléguée aux agents composant le SMA.

Depuis sa création à la fin des années 60 jusqu'à aujourd'hui, l'IAD a connu une évolution importante, et les SMA ont bénéficié de nombreuses améliorations, tant au niveau de leur cadre de conception que dans la compréhension de leurs dynamiques.

Ainsi, les années 90 ont intégré aux SMA une somme de connaissances issues de disciplines diverses telles que la biologie et la sociologie. Les caractéristiques d'auto-organisation, d'émergence et d'interactions entre agents ont ainsi été précisément définies, et leurs impacts dans la résolution des problèmes à l'aide de SMA ont été étudiés. En particulier, les travaux de l'équipe SMAC de l'IRIT, au sein de laquelle se déroule cette thèse, se sont focalisés sur la résolution

de problèmes complexes grâce à l'utilisation de SMA centrés sur la notion d'émergence.

L'utilisation des SMA a rapidement permis d'offrir des solutions nouvelles à des problèmes complexes. En effet, la décomposition de ces problèmes en plusieurs sous-problèmes plus facilement traitables a contribué à populariser l'emploi des SMA dans les cas où la solution globale d'un problème est difficilement accessible algorithmiquement, ce qui est le cas du contrôle de systèmes dynamiques.

Les paragraphes ci-après détaillent la notion d'agent et les différents types d'agents existant. Ces définitions permettent d'introduire les mécanismes régissant un SMA et, en particulier, son organisation.

Le chapitre suivant, consacré à l'état de l'art, s'appuiera sur les notions présentées ici pour exposer différentes approches utilisant des SMA pour contrôler des procédés complexes.

II.2.1 La notion d'agent

Les agents forment les éléments de base d'un Système Multi-Agent (SMA). Il est donc nécessaire de les définir et d'insister sur les différentes utilisations qu'il est possible d'en faire. Les sections suivantes introduiront les concepts permettant de comprendre le rôle d'un agent, ainsi qu'une description des différentes formes qu'il peut adopter.

II.2.1.1 Définition d'un agent

D'un point de vue informatique, la notion d'agent est apparue comme une évolution des objets, et s'est peu à peu imposée comme une alternative possible en tant que fondement d'un nouveau paradigme de programmation : la programmation orientée agent.

Cette approche de la programmation bénéficia de parallèles forts la connectant aux sciences sociales ainsi qu'aux phénomènes biologiques et connut un rapide essor grâce à cet aspect multidisciplinaire.

Même si cela peut paraître étrange, il existe de nombreuses définitions décrivant ce qu'est un agent. Ces définitions sont relativement proches les unes des autres, mais chacune accorde une importance particulière à différents points, offrant ainsi un éclairage particulier sur certaines notions clés.

L'une des propositions les plus répandues est donnée par [Ferber 1995], qui définit un agent de la manière suivante :

« On appelle agent une entité physique ou virtuelle

- qui est capable d'agir dans un environnement,
- qui peut communiquer directement avec d'autres agents,
- qui est mue par un ensemble de tendances (sous la forme d'objectifs individuels ou d'une fonction de satisfaction, voire de survie, qu'elle cherche à optimiser),
- qui possède des ressources propres,
- qui est capable de percevoir (mais de manière limitée) son environnement,
- qui ne dispose que d'une représentation partielle de cet environnement (et éventuellement aucune),
- qui possède des compétences et offre des services,
- qui peut éventuellement se reproduire,
- dont le comportement tend à satisfaire ses objectifs, en tenant compte des ressources et des compétences dont elle dispose et en fonction de sa perception, de ses représentations et des communications qu'elle reçoit. »

Parmi l'ensemble de ces caractéristiques, certaines peuvent être regroupées sous une nomenclature plus précise. En ce sens, [Wooldridge 1995] détaille un agent comme :

« ... a hardware or (more usually) software-based computer system that enjoys the following properties :

- autonomy : agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state ;
- social ability : agents interact with other agents (and possibly humans) via some kind of agent-communication language ;
- reactivity : agents perceive their environment, (which may be the physical world, a user via a graphical user interface, a collection of other agents, the Internet, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it ;
- pro-activeness : agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking the initiative. »

Cette seconde définition explicite la notion d'autonomie et fait intervenir la dimension sociale existant entre les agents eux-mêmes. Cette dimension sociale inclut également l'utilisateur, qui peut donc communiquer avec les agents, par exemple, par l'envoi de messages.

L'ensemble de ces définitions permet de définir un agent selon les quatre axes suivants :

- Le premier repose sur la place de l'autonomie en tant que caractéristique principale d'un agent, caractéristique le différenciant d'un objet. L'agent décide de ses actions, ce n'est

pas une entité passive agissant seulement lorsqu'elle reçoit un message. Cette autonomie est la caractéristique la plus difficile à appréhender lorsque l'on travaille avec des agents : en effet, certains agents peuvent posséder une autonomie très limitée, semblant remettre en cause la nature d'agent des composants du système. Il est donc nécessaire de noter qu'un objet, contrairement à un agent, n'encapsule pas le choix de ses actions.

– L'aspect local d'un agent apparaît dans un second temps. Un agent agit selon ses propres objectifs, ses perceptions et ses connaissances locales. Ce dernier point préfigure l'un des avantages majeurs d'un système multi-agent, à savoir que les agents le composant n'ont pas besoin de connaître le problème global à résoudre (ni par extension, les moyens de le résoudre), tout en pouvant agir localement et contribuer à sa résolution.
– La troisième caractéristique est le caractère social d'un agent. Les définitions données précédemment pour les agents soulignent leurs capacités de communication, ainsi que les interactions auxquelles ils sont sujets. Un point essentiel à dériver de cette idée est qu'un agent est donc capable de communiquer avec d'autres agents de nature différente. De ce fait, un système peut être composé d'une multitude d'agents hétérogènes, capables d'interagir les uns avec les autres.
– Le dernier axe est formé par la notion d'environnement. Un agent est donc une entité située à l'intérieur d'un environnement qu'il peut percevoir et sur lequel il peut agir. Cette double compétence, perception et action, est à l'origine des capacités d'adaptation de l'agent, lui permettant de modifier son environnement selon ses objectifs propres et de réagir en fonction des modifications qu'il perçoit.

Il est à noter que parmi les différentes définitions d'un agent, [Luck 2005] propose une alternative, à savoir « '... a design metaphor ». Ici, un agent est une entité autonome et communicante servant de base à la réalisation d'applications immergées dans un environnement fortement dynamique. Cette définition s'adapte particulièrement à la notion d'architecture orientée service et, comparée à [Ferber 1995, Wooldridge 1995], fournit une vision à une échelle plus élevée, d'entités en nombre plus limité et au niveau de cognition supérieur.

Les définitions d'un agent sont donc variées tout en restant relativement génériques et cela impose de caractériser plus finement les agents afin de les classer en différents groupes pour décrire les SMA les utilisant.

Tableau II.1 – Les différents types d'agents

Type d'agents	Réactif	Cognitif
Environnement	Représentation implicite	Représentation explicite
Apprentissage	Pas ou peu de mémoire	Peut utiliser des méthodes d'apprentissage complexes
Quantité d'agents	Importante	Limitée

II.2.1.2 Les différents types d'agents

Les différentes définitions du terme agent conduisent à l'apparition d'une multitude de types d'agents distincts dont la classification précise est sujette à débat. En effet, les agents sont un paradigme de programmation à mi-chemin entre une abstraction générique et une technique d'application particulière. Parmi les distinctions réalisables, il est commun de catégoriser les agents selon leur niveau de cognition. Deux familles principales se distinguent alors, les agents dits cognitifs et les agents dits réactifs.

Un agent cognitif possède une base de connaissance lui permettant de raisonner sur son environnement, ses interactions et de définir des objectifs qui lui sont propres. Ainsi, un agent cognitif est capable d'apprendre de ses expériences et de planifier ses actions en vue de satisfaire son objectif propre.

Un agent réactif pour sa part apparaît comme moins « intelligent ». Il se contente de réagir à l'aide de ses perceptions selon un comportement prédéfini et ne possède que rarement une mémoire des évènements passés. L'intérêt de tels agents est dans le nombre, car à puissance de calcul égale et du fait de la simplicité de leurs comportements, un système multi-agent est capable de gérer une quantité d'agents réactifs plus importante que celle d'agents cognitifs.

De ce fait, un système composé d'agents réactifs se base sur les interactions nombreuses entre une grande quantité d'agents, tandis qu'un système utilisant des agents cognitifs se concentrera sur des communications moins nombreuses, mais potentiellement plus riches telles que des négociations.

Le tableau II.1 résume ces caractéristiques.

La présence d'agents réactifs peut brouiller la frontière existant entre les agents et les objets, il convient alors de garder en mémoire que les aspects sociaux des agents ainsi que la granularité de ces deux éléments forment au final des concepts bien distincts. Ainsi, même s'il n'existe pas de caractérisation absolue de ce qui est ou non un agent, les entités réactives présentées ici sont

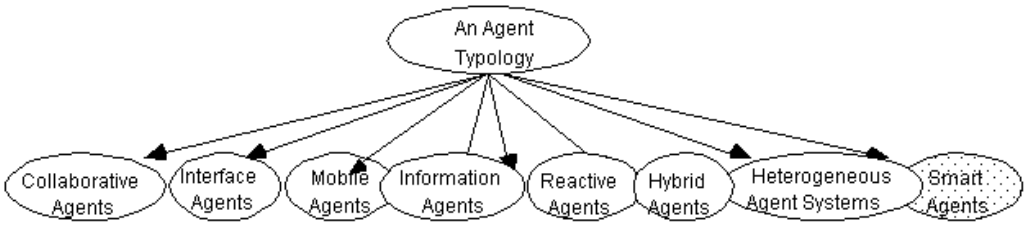

Figure II.4 – Différents types d'agents d'après [Nwana 1996]

considérées comme des agents.

Au-delà de cette catégorisation, il est possible de classer les agents selon leur architecture interne, qu'elle soit dirigée par un modèle, conduite par des objectifs, par une heuristique ou centrée sur l'apprentissage.

Par exemple, [Nwana 1996] présente une nouvelle différenciation des types d'agents (voir Figure II.4). Cette catégorisation tient donc compte d'une définition plus fine de la notion d'agent et s'attache à les regrouper selon leurs architectures internes et leurs objectifs. Il n'est alors plus question des capacités cognitives d'un agent, mais d'une distinction préfigurant le rôle que jouera l'agent au sein du système à l'intérieur duquel il évoluera.

Un agent est donc une entité pouvant jouer un grand nombre de rôles différents, et par là même, servir de base à quantité de systèmes aux fonctionnements variés. Une première information sur l'organisation et le fonctionnement d'un SMA peut donc être dérivée de la complexité des agents le composant, ainsi que de leur architecture interne.

II.2.2 Les interactions entre agents

Les définitions d'un agent exprimées dans la partie II.2.1.1, nous permettent de distinguer la présence de capacités de communications entre agents, ainsi que les premières mentions de sociétés d'agents.

Cette communication sous-entend la présence de plusieurs agents capables d'interagir entre eux au sein du même système et donc d'une hétérogénéité potentielle.

Cependant, la création d'un SMA ne se résume pas à la définition d'une collection d'agents, mais nécessite une description précise de leurs interactions comme le développe la partie suivante. Les mécanismes régissant l'organisation d'un SMA ou lui permettant de s'auto-organiser pour résoudre un problème sont donc un élément essentiel à considérer.

II.2.2.1 Composition d'un SMA

Afin de former un SMA, un ensemble d'éléments doivent être réunis. La définition la plus synthétique est fournie par [Erceau 1991], pour qui un SMA est un système <O, E, A> où : O est un ensemble d'objets, A est un ensemble composé d'agents, O et A étant immergés dans un environnement E.

Une définition sensiblement différente est fournie par l'approche Voyelles [Demazeau 1995], proposant une décomposition d'un SMA en quatre composants élémentaires :

- Agents : agents incluant l'ensemble des éléments composant les modèles (ou les architectures) utilisés pour décrire leurs comportements. Ceux-ci peuvent être d'une complexité quelconque, d'un système réactif à un agent raisonnant à partir d'une base de connaissance.
- Environnement : élément décrivant le milieu dans lequel sont plongés les agents. L'environnement rejoint la définition fournie dans la partie II.2.1.1 et ne se limite pas forcément à un positionnement spatial.
- Interactions : éléments concernant l'ensemble des moyens mis en œuvre afin de gérer les interactions entre les agents du SMA, en particulier les protocoles de communications mis en place, ainsi que la gestion des interactions physiques.
- Organisations : éléments permettant la structuration d'un ensemble d'agents, sous la forme d'entités complexes telles que des hiérarchies ou des groupes sociaux, ou de simples relations.

L'intérêt de cette approche repose sur le principe de récursion. En effet, chaque agent d'un système, composé selon l'approche Voyelle, peut être à son tour considéré comme un SMA, offrant ainsi une grande généricité à cette approche.

Au-delà de la notion d'agent définie dans la partie II.2.1.1 et de celle d'environnement fortement dépendante du domaine à traiter, les éléments d'interaction et d'organisation restent à détailler afin de pouvoir établir un panorama des SMA permettant de juger de la pertinence de leur application au problème du contrôle de bioprocédés.

II.2.2.2 L'organisation des agents

Présenter la diversité des différentes approches multi-agents existantes passe par la description des différentes organisations des agents. Celles-ci offrent une autonomie variable aux agents du système et de ce fait, contraignent plus ou moins leurs comportements. En contrepar-

tie, l'utilisation d'une organisation particulière permet de s'assurer que le système développé respectera une certaine cohérence vis-à-vis de son comportement global.

Les sections suivantes détaillent les organisations des SMA les plus couramment utilisées, et précisent leur impact sur le comportement des agents ainsi que sur l'évolution d'un SMA dans sa recherche de la solution au problème qu'il doit traiter.

II.2.2.2.1 Organisation AGR/AGRE La première organisation d'agents présentée est nommée AGR pour Agent-Groupe-Rôle. Cette organisation est définie dans [Ferber 2003] et repose sur les composants suivants :

- Agent : un agent est une entité active et communicante, jouant un rôle au sein d'un groupe. Un agent peut assumer plusieurs rôles et appartenir à de nombreux groupes différents. Aucun présupposé sur le niveau de cognition de l'agent n'est effectué ici, ce modèle permettant donc l'utilisation d'agents réactifs ou cognitifs. De plus, l'architecture interne de l'agent reste à l'appréciation du concepteur.
- Groupe : un groupe est un ensemble d'agents partageant des caractéristiques communes. Cette entité structure donc les agents par regroupement contextuel selon les activités à accomplir. En effet, deux agents peuvent communiquer entre eux si et seulement si ils appartiennent à un groupe commun.
- Rôle : un rôle est une représentation abstraite de la fonction d'un agent au sein d'un groupe. Un agent doit jouer au moins un rôle dans le groupe, rôles locaux aux groupes. Un rôle peut être joué par plusieurs agents.

Des travaux récents ont abouti à des extensions du modèle AGR : AGRE [Ferber 2005] prend en compte l'environnement dans la modélisation et AGREEN [Baez 2005] tente d'intégrer à la fois la dimension environnementale et normative des organisations multi-agents. Les normes de AGREEN sont construites autour d'opérateurs déontiques et définissent des sanctions applicables en cas de leur non-respect.

Cet exemple d'organisation AGR est l'un des plus répandu de par sa généricité et sa simplicité. Il complète naturellement la définition d'un SMA, sans ajouter de couche d'abstraction théorique pour décrire les relations entre agents.

L'organisation AGR se trouve donc naturellement adaptée à l'application d'un SMA à un problème pour lequel les rôles des différents agents sont spécifiables. De surcroît, cette approche peut être instanciée à l'aide d'une large gamme d'agents de types distincts, aucun présupposé n'étant fait sur leurs comportements ou sur leur niveau de cognition requis.

II.2.2.2.2 Organisation holonique Le terme holon est avant tout un concept philosophique. Il a, pour la première fois, été employé par [Koestler 1979] et désigne une entité auto-similaire, stable, cohérente et autonome, composée de plusieurs holons ou elle-même faisant partie d'un tout d'ordre supérieur. Cette imbrication de structures confère aux holons une place particulière. En effet, les actions d'un holon doivent satisfaire ses propres objectifs tout en favorisant l'atteinte des objectifs du holon dont il fait partie. Ainsi, un holon possède un comportement combinant autonomie et coopération.

Ce point marque la principale différence entre un holon et un agent : un holon doit posséder un comportement coopératif, tandis qu'un agent ne possède *a priori* aucune contrainte quant au comportement qu'il doit adopter. Un système centré sur l'utilisation de holons peut donc être considéré comme équivalent à un SMA dont le comportement des agents vérifie un ensemble de contraintes de coopération.

Une structure composée de holons forme une holarchie, telle que représentée par la figure II.5.

Une holarchie est au final plus souple qu'une hiérarchie grâce à la communication entre holons appartenant à un même niveau. Cette organisation peut en effet évoluer dynamiquement selon le comportement des holons : une organisation holonique peut être auto-organisée grâce à la coopération entre holons, permettant d'inclure de nouveaux holons dans le système ou, à l'inverse, de supprimer ceux n'agissant pas dans l'intérêt global.

Son utilisation en tant qu'organisation d'un SMA permet de fournir différents niveaux d'abstractions, de nombreux points de vue sur le système ainsi que le maintien d'interactions locales entre un grand nombre d'entités.

Ce type d'organisation présente donc une structure équilibrée entre autonomie et ouverture, au prix de la nécessité d'intégrer un comportement coopératif aux agents la composant. Des applications concrètes basées sur cette approche ont par exemple permis de traiter des problèmes de transport routier [Versteegh 2010] ou encore de gestion de bagages [Black 2007].

II.2.2.2.3 Auto-organisation L'auto-organisation est avant tout une notion physique et biologique, concernant l'état de certains systèmes capables d'exhiber des fonctions ou des formes structurées sans intervention extérieure. Ce phénomène concerne également des populations d'individus et fait donc également l'objet de nombreux travaux en sciences sociales.

Ainsi, des phénomènes aussi différents que la percolation, les cristaux liquides, ou la formation d'étoiles et de galaxies font intervenir une auto-organisation des composants impliqués.

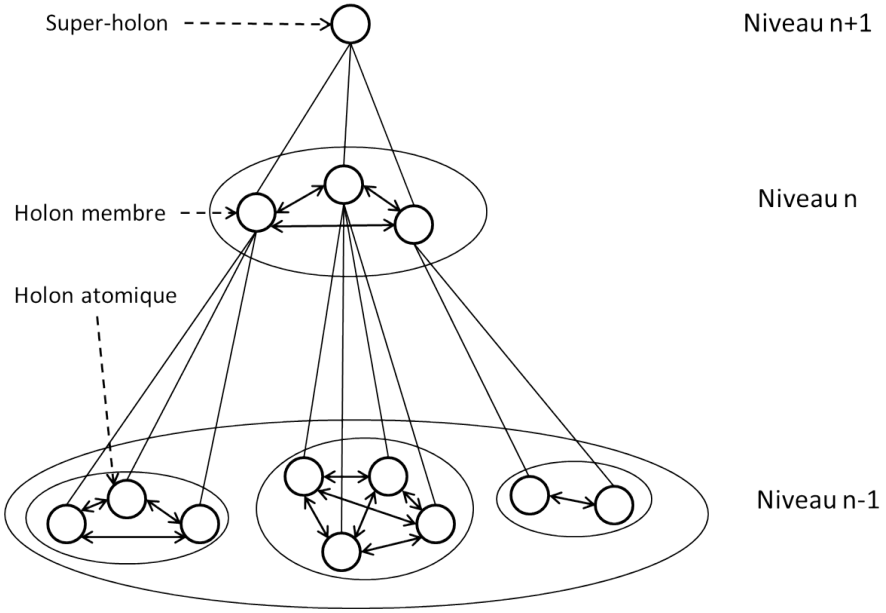

Figure II.5 – Représentation d'une holarchie

D'un point de vue informatique, l'auto-organisation peut être considérée comme le mécanisme permettant à un système de modifier son organisation durant son exécution sans commande externe explicite [Serugendo 2005]. De ce fait, seules les informations locales aux agents sont utilisées afin de définir la structure globale du système. Parmi celles-ci, nous pouvons citer le rôle essentiel de l'environnement, par le biais de la stigmergie, étudié dans [Hassas 2003]. Cette notion de stigmergie décrit un mécanisme d'auto-structuration de l'environnement lié aux actions des agents. L'évolution de l'environnement est perçue par les agents, influençant ainsi leur comportement et de fait, leurs actions sur l'environnement. Cet ensemble d'interactions entre les agents et l'environnement conduit à l'auto-organisation de l'ensemble du système.

Il existe de nombreuses méthodes permettant d'appliquer cette auto-organisation à une population d'agent, chacune d'entre elles étant inspirée par un domaine différent.

Par exemple, les comportements des bancs de poissons ou des nuées d'oiseaux ont inspiré les différentes techniques d'intelligence par essaims [Kennedy 2006]. Celles-ci présentent des agents réactifs aux comportements simples et dont l'auto-organisation conduit au comportement global du système.

Considérons le cas de l'algorithme de colonies de fourmis décrit par [Dorigo 2006]. Le principe de base d'une telle organisation repose sur l'observation de fourmis recherchant de la nourriture. Ces fourmis se déplacent par défaut de manière aléatoire jusqu'à trouver une source de nourriture. Une fois la nourriture repérée, les fourmis retournent vers la fourmilière en déposant

des phéromones que les autres fourmis seront tentées de suivre. Ces phéromones s'évaporent au cours du temps, conduisant ainsi la quantité de phéromones présente sur les chemins les plus empruntés à rester élevée, contrairement à celle présente sur des chemins moins optimisés car plus long. Ainsi, l'organisation des fourmis se modifie peu à peu vers le chemin le plus court entre la fourmilière et la source de nourriture.

Ce type de comportement appliqué aux agents d'un SMA a permis de traiter des problèmes tels que le repliement des protéines [Shmygelska 2005] ou le routage de télécommunication [Di Caro 2004].

Un autre exemple, cette fois-ci inspiré des sciences sociales, est fourni par les mécanismes de propagation de rumeurs, source de l'algorithme T-man [Jelasity 2006]. Un système dont l'organisation est centrée sur la propagation de rumeurs fonctionne de la manière suivante : chaque agent du système possède une vue locale de ses agents-voisins proches, ainsi qu'une fonction de classement lui permettant de réorganiser sa liste de voisins. Chaque agent va alors transmettre à l'agent le plus proche, selon sa fonction de classement, la liste des agents qu'il connaît, ainsi que les informations les concernant. Ces nouvelles données permettent à l'agent qui les reçoit de mettre à jour son voisinage à l'aide de sa propre fonction de classement et ainsi, d'éventuellement modifier l'agent dont il est le plus proche. Le résultat d'une telle approche est une réorganisation possible du SMA, conférant au système une grande robustesse quant à l'ajout ou la suppression d'agents.

Ainsi, cette méthode d'auto-organisation est particulièrement adaptée à la gestion de la topologie d'un réseau et aux systèmes pair-à-pair par exemple.

Enfin, comme mentionné dans la section II.2.2.2.2, des systèmes holoniques peuvent également être auto-organisés. Dans ce cas, la coopération guide l'auto-organisation et permet au SMA de s'adapter à l'ajout ou la suppression d'agents et d'ainsi modifier sa propre structure afin d'atteindre son objectif.

L'auto-organisation présente un intérêt certain, en déléguant les contraintes d'organisation du SMA aux agents eux-mêmes. La complexité de cette étape se retrouve donc, de manière différente, au sein du comportement de chacun des agents et de ce fait, influe sur la manière de les développer afin d'assurer cette auto-organisation. Cette approche est donc particulièrement adaptée lorsqu'il est impossible de définir une organisation appropriée de par la complexité du SMA à organiser. Le travail du concepteur du système se focalise alors sur les comportements locaux des agents, et l'organisation globale du SMA émergera de l'ensemble de ces comportements.

II.2.3 Bilan des SMA

Pour conclure cette présentation des SMA, on peut ajouter que les systèmes multi-agents ont été utilisés pour résoudre une multitude de problèmes distincts partageant toutefois des caractéristiques particulières qui sont les suivantes selon [Parunak 2000] :

- Modulaires : les applications modulaires, c'est-à-dire dont les différentes entités les composant forment un découpage précis, facilitent l'application d'un SMA. En effet, la proximité des agents avec les objets au sens informatique permet de s'adapter à une décomposition existante.
- Décentralisées : point proche du côté modulaire, l'aspect décentralisé comporte en plus une notion d'autonomie, qui peut être reliée facilement aux capacités de prises de décisions locales d'un agent appartenant à un SMA.
- Dynamiques : l'intérêt de l'utilisation d'un SMA pour traiter une problématique fortement dynamique dérive des deux points précédents. En effet, une application connaissant des modifications fréquentes et rapides bénéficiera d'autant plus des avantages de prises de décisions locales et de la robustesse liée à une conception modulaire.
- Non-entièrement observables : cette caractéristique concerne les applications pour lesquelles des résultats doivent être produits même lorsque toutes les données ne sont pas accessibles. Dans un SMA, un agent ne possédant pas toutes les informations attendues ne bloque pas le fonctionnement global du système.
- Complexes : enfin, les SMA s'adaptent particulièrement aux applications complexes, faisant intervenir une grande quantité de comportements différents et un nombre élevé d'interactions.

De plus, les multiples organisations possibles quant aux interactions entre agents permettent de définir une multitude de systèmes, s'étendant de la définition d'un essaim d'agents réactifs s'auto-organisant en fonction de leur environnement, à l'ensemble d'agents cognitifs spécialisés organisés en une hiérarchie fixe.

Parmi ces organisations, les facultés d'auto-organisation d'un SMA, dirigeant l'adaptation de sa structure à la résolution d'un problème donné, permettent d'envisager la réalisation d'un SMA de contrôle dont l'évolution de l'organisation des agents le composant donnerait lieu à un comportement global adapté à la dynamique du système qu'il doit contrôler.

En résumé, les SMA permettent de traiter des problèmes complexes en s'appuyant, d'une part, sur les capacités de résolution locale des agents et, d'autre part, sur les capacités d'adaptation issues de leurs interactions. Leur application à des problèmes complexes tels que le contrôle

de procédés est donc naturelle.

II.3 Bilan

Pour situer le contexte dans lequel se déroule ce travail de thèse, ce chapitre a d'abord défini le problème du contrôle de systèmes ainsi que les notions associées, en fonction du domaine d'application visé. Contrôler un système dynamique, tel qu'un bioprocédé, par exemple, peut être qualifié de problème complexe : la nature du système à contrôler peut grandement varier, sa dynamique est généralement changeante et des non-linéarités existent. La solution proposée doit donc posséder des propriétés d'auto-adaptation. Tout d'abord, pour être capable de s'adapter à la nature différente des systèmes à contrôler et d'alléger ainsi le travail d'adaptation requis pour l'appliquer à un procédé en particulier. Pour s'adapter ensuite à la dynamique changeante du système à contrôler et posséder ainsi la robustesse lui permettant de satisfaire les objectifs de l'utilisateur.

Dans un second temps, ce chapitre a défini les concepts principaux liés au domaine des systèmes multi-agents qui peuvent s'avérer une technologie potentiellement génératrice d'une solution au problème ici posé. En étudiant les caractéristiques principales de ces systèmes, on peut conclure que les SMA constituent une technologie adéquate pour résoudre des problèmes complexes tels que le contrôle de systèmes dynamiques.

Bien entendu, des solutions au contrôle sont déjà envisageables et peuvent s'appuyer sur des techniques issues de domaines aussi différents que les mathématiques ou l'intelligence artificielle. Le chapitre suivant a pour objet d'étudier et de comparer ces solutions pour déterminer leur adéquation ou leurs limites pour le contrôle de bioprocédés tel qu'envisagé dans ce travail.

Chapitre III

Etat de l'art

Une fois le contexte de ce travail de thèse établi, il est possible de dresser un panorama des méthodes habituellement employées afin de contrôler des systèmes dynamiques provenant ou non du domaine de la biologie. Ces techniques, issues de domaines variés, offrent une vue d'ensemble des différents « contrôles » existant. L'objectif de ce chapitre est de comparer leurs avantages et leurs limites en considérant leur application au contrôle de systèmes ayant des propriétés telles que les bioprocédés cibles de ce travail.

La description de ces techniques de contrôle commence par celle des régulateurs PID (Proportional - Integral - Derivative) qui sont l'une des approches les plus classiques et les plus utilisées dans l'industrie. Différentes techniques de contrôle adaptatif, plus adaptées au problème fortement non linéaire que nous devons résoudre, sont ensuite présentées. La troisième partie étudie les techniques de contrôle dites « intelligentes », issues de l'intelligence artificielle. Pour terminer cet état de l'art, une partie est plus spécifiquement consacrée aux approches mettant en jeu des systèmes multi-agents.

De manière générale, la description suit un ordre de complexité croissant, en partant des approches locales centrées sur la régulation d'une variable unique jusqu'aux systèmes de supervision modélisant le système à contrôler.

III.1 Les régulateurs PID

Dans l'industrie, et en particulier en biologie, le système de contrôle classique lié aux rétroactions le plus utilisé est nommé Régulateur PID, signifiant Régulateur Proportional-Integral-Derivative [Aastrom 1995], du nom des trois fonctions mathématiques qu'il applique. Ce type

de contrôleur permet de traiter le signal de rétroaction reçu afin de déterminer le contrôle à effectuer. Ces trois modes de fonctionnement vont donc chacun calculer une valeur liée à la rétroaction reçue, et une combinaison de ces trois valeurs déterminera le contrôle à effectuer sur le système. Il est à noter que selon les besoins, les fonctions « Proportional » et « Integral » peuvent être appliquées sans la fonction « Derivative » ; nous obtenons alors un régulateur de type PI. De manière similaire, nous parlerons de régulateur P lorsque seule la fonction « Proportional » est appliquée, ou de régulateur PD pour l'application des fonctions « Proportional » et « Derivative ».

Concrètement, l'action réalisée par les différentes fonctions est la suivante :

- Le terme « Proportional », également appelé « gain », permet de multiplier l'erreur actuelle mesurée par une « constante proportionnelle ».
  $P = K_p e(t)$
  où K_p est la constante proportionnelle, e la différence entre la valeur mesurée et la valeur idéale attendue et t le temps.
- Le terme « Integral » permet de considérer la durée de l'erreur en plus de son amplitude. Ainsi, il tient compte de la somme des erreurs accumulées et multiplie ce résultat par une « constante intégrale ».
  $I = K_i \int_0^t e(\theta)\, d\theta$
  où K_i est la constante intégrale, e la différence entre la valeur mesurée et la valeur idéale attendue, t le temps et θ la variable d'intégration.
- Le terme « Derivative » offre quant à lui la possibilité d'estimer le taux de variation des erreurs rencontrées et ainsi d'observer si les erreurs augmentent ou diminuent. Ici aussi, le résultat est multiplié par une constante dite « constante dérivée ». Ainsi, cette fonction permet une accélération de la correction lorsque l'erreur mesurée s'accroît brutalement.
  $D = K_d d/dt e(t)$
  où K_d est la constante dérivée, e la différence entre la valeur mesurée et la valeur idéale attendue et t le temps,

Au final, ces trois termes permettent d'obtenir le résultat selon une combinaison propre au régulateur. Le plus souvent, cette combinaison est une somme, conduisant donc à la formule suivante pour P, I et D.

$M = P + I + D$

Il reste toutefois nécessaire de définir la manière dont chacune de ces fonctions sera appliquée, ainsi que leurs paramètres, en particulier la valeur des différentes constantes. Cette étape

Tableau III.1 – Méthode de Ziegler-Nichols pour l'ajustement des paramètres d'un régulateur PID

Control	K_c	T_I	T_D
P	$K_u/2$		
PI	$K_u/2.2$	$P_u/1.2$	
PID	$K_u/1.7$	$P_u/2$	$P_u/8$

constitue le réglage d'un PID. Afin de permettre à ces paramètres de voir leur valeur modifiée au cours du temps, et ainsi de mieux s'adapter à l'état du bioprocédé, plusieurs méthodes existent. Parmi elles, et sans compter l'ajustement manuel possible, les principales règles définissant cet ajustement sont celles de Ziegler-Nichols et Cohen-Coon.

III.1.1 Méthode de Ziegler-Nichols

La méthode de Ziegler-Nichols est une heuristique, issue de [Ziegler 1942], permettant l'ajustement des constantes du régulateur PID selon la table III.1. L'idée est ici d'annuler les gains des fonctions I et D, tout en augmentant celui de P, jusqu'à ce que sa valeur atteigne le gain critique à partir duquel la sortie de la boucle de contrôle oscille de manière périodique. Ce gain critique, noté K_u, et la durée de ces oscillations, notée Pu, calculée grâce à la formule $P_u = 2p/w_c$, permettent alors d'ajuster les constantes des formules de calcul de P, I et D à l'aide des coefficients du tableau.

Bien qu'elle ne soit pas optimale, cette méthode demeure très solide et continue d'être appliquée depuis les années 1940. Cependant, elle reste relativement agressive dans le sens où lors de l'ajustement du P, elle peut entraîner le système vers un état instable.

Ainsi, de nombreuses extensions, ou améliorations, y ont été apportées, telles que la méthode Tyreus-Luyben [Tyreus 1992] modifiant les valeurs du tableau d'ajustement utilisé dans Ziegler-Nichols.

III.1.2 Méthode de Cohen-Coon

Cette méthode suit le même objectif que celle de Ziegler-Nichols, la différence principale réside dans la manière dont les constantes des différentes formules du régulateur PID sont calculées. De plus, la méthode de Cohen-Coon est dite hors-ligne, contrairement à celle de Ziegler-Nichols. En effet, les modifications doivent être effectuées une fois que le système a

Tableau III.2 – Méthode de Cohen-Coon pour l'ajustement des paramètres d'un régulateur PID

Control	K_c	τ_I	τ_D
PI	$\frac{1}{k}\frac{\tau}{\theta}\left[0.9+(\frac{\theta}{12})\tau\right]$	$\frac{\theta\left[30+(\frac{3\theta}{\tau})\right]}{9+20\frac{\theta}{\tau}}$	
PID	$\frac{1}{k}\frac{\tau}{\theta}\left[\frac{16\tau+3\theta}{12\tau}\right]$	$\frac{\theta\left[32+(\frac{6\theta}{\tau})\right]}{13+8\frac{\theta}{\tau}}$	$\frac{4\theta}{11+2\frac{\theta}{\tau}}$

atteint un état stable.

Les valeurs fournies par Cohen-Coon sont détaillées dans le tableau III.2, où τ est la constante de temps du procédé, θ son « dead-time » et k son gain.

Cette méthode est essentiellement utilisée sur des procédés présentant des « dead-times » importants et ne s'applique qu'aux modèles du premier ordre. Elle offre une réponse rapide mais souffre des nombreuses contraintes qu'elle impose quant à son application car elle ne s'applique que sur un nombre limité de modèles. De plus, les résultats obtenus sur des systèmes dont les « dead-times » sont faibles tendent à être de moins bonne qualité que ceux fournis par la méthode de Ziegler-Nichols. L'inverse se produit lorsque les « dead-times » deviennent plus importants.

III.1.3 Avantages et limites

L'avantage des régulateurs PID réside dans leur relative généricité, car bien que nécessitant un ajustement de leurs variables dépendant du problème, ils demeurent applicables à un grand nombre de systèmes de contrôle, faisant de cette approche la plus répandue actuellement.

Cependant, de tels systèmes peuvent se révéler coûteux à mettre en œuvre et nécessitent des réajustements fréquents au niveau matériel afin d'assurer la stabilité des mesures. Cela est dû à la place des régulateurs PID dans les systèmes de contrôle de bioprocédés. En effet, les régulateurs PID se situent essentiellement au niveau du contrôle d'une variable définie, telle que la température ou le pH par exemple, et à partir des mesures effectuées, ces régulateurs décident des modifications à y apporter. La difficulté réside dans le fait que les éléments physiques permettant de mesurer les valeurs nécessaires au bon fonctionnement du régulateur, et d'appliquer les modifications sur le bioprocédé, ne doivent en aucun cas perturber le déroulement de celui-ci, ce qui implique les contraintes matérielles mentionnées.

De plus, les performances des régulateurs PID dans les systèmes non-linéaires sont variables, ce point étant particulièrement apparent si l'on conserve l'algorithme de base. De ce fait, il est

Tableau III.3 – Récapitulatif des caractéristiques du PID

Nom du contrôle / Critère de comparaison	PID
Adaptabilité	- (nécessite de lourdes modifications)
Généricité	+
Position	Locale (niveau d'une variable unique)
Remarques	Nécessite des améliorations complexes pour s'adapter à des systèmes non-linéaires

nécessaire d'adapter le type de contrôle utilisé en lui adjoignant, par exemple, des mécanismes de raisonnement reposant sur la logique floue [Visioli 2001], l'utilisation de régulateurs PID en cascade [Lee 1998] ou d'autres mécanismes utilisant des connaissances extérieures sur le système à contrôler (feed-forward control).

Enfin, une des limites de l'utilisation de régulateurs PID provient de leur incapacité à traiter plusieurs variables tout en considérant plusieurs rétroactions. Le bloc de contrôle ainsi formé ne concerne donc qu'une variable ciblée, et le contrôle appliqué ne possède donc que la vue strictement locale de la rétroaction qu'il reçoit. Les différentes caractéristiques de ce type de contrôle au niveau de son adaptabilité, de sa généricité, et de sa complexité sont résumées dans le tableau III.3.

Les limites énoncées des approches du contrôle basées sur les régulateurs PID nous ont donc conduits à étudier les systèmes de contrôle adaptatif.

III.2 Le contrôle adaptatif

Nos travaux portant sur des systèmes fortement non linéaires, nous nous sommes particulièrement intéressés à l'utilisation du contrôle adaptatif. En effet, les résultats obtenus à partir de mesures sur un bioprocédé peuvent varier grandement d'une itération du bioprocédé à l'autre, entraînant ainsi des conséquences non négligeables concernant son optimisation et son contrôle, telles que l'ajout de bruit et de délais de réaction variables. Afin de répondre à cette difficulté, le contrôle adaptatif propose donc des mécanismes de modification de la loi de contrôle du système tenant compte de cet aspect dynamique. Ces techniques reposent principalement sur l'exploitation ou la création d'un modèle du système à contrôler.

Lorsque nous considérons le problème du contrôle de systèmes complexes en général, les méthodes de contrôle adaptatif peuvent être divisées en trois grandes familles, basées sur l'identification de modèle (Model Identification Adaptive Control (MIAC)) [Soderstrom 1988], l'utilisation d'un modèle de référence (Model Reference Adaptive Control (MRAC)) [H.P. Whitaker 1958] ou l'utilisation de contrôles spécifiques à l'obtention d'informations sur le système à contrôler (Dual Control Theory) [Feldbaum 1961].

III.2.1 Systèmes MIAC

Les systèmes MIAC utilisent des mécanismes d'identification de modèles, permettant au contrôleur de se forger un modèle du procédé à contrôler. Ce modèle peut être entièrement généré selon les observations effectuées ou construit à partir d'une base connue, et sa complexité est variable selon les méthodes de modélisation employées. Ainsi, le contrôleur utilisera les informations fournies par ce système d'identification afin d'effectuer les actions de contrôle nécessaires. Ce principe de fonctionnement est détaillé dans la figure III.1. Il est à noter que le système permettant l'identification du modèle est relié à la fois aux entrées et aux sorties du système à contrôler, afin de déterminer précisément le modèle à utiliser et de transmettre cette information aux mécanismes d'ajustement du contrôleur.

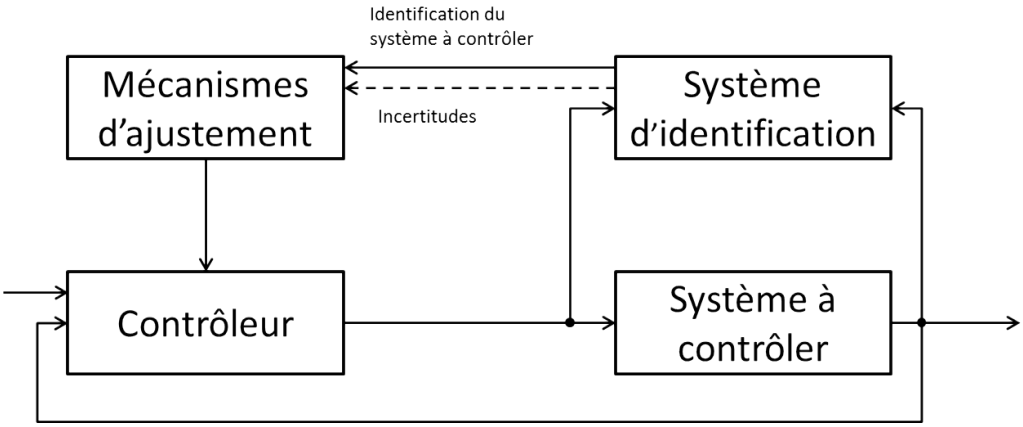

Figure III.1 – Fonctionnement d'un MIAC

III.2.2 Systèmes MRAC

Les systèmes MRAC, suggérés par Whitaker au MIT en 1958, tels que décrits sur la figure III.2, ont pour objectif de créer une boucle fermée dont les paramètres peuvent être ajustés

selon le (ou les) modèle(s) utilisé(s), une loi de contrôle étant calculée à partir de ce modèle de référence. Le mécanisme d'ajustement utilise donc les connaissances contenues dans ce modèle, ainsi que les observations effectuées sur les entrées et les sorties, afin d'ajuster les paramètres du contrôleur selon l'écart entre les prévisions et les résultats obtenus.

Cet ajustement est généralement réalisé à l'aide de la « MIT rule » [Astrom 1994] qui est un type de processus de descente en gradient cherchant à minimiser un critère de performance représenté par l'intégrale du carré de l'erreur mesurée. Cette erreur est l'écart entre la valeur de sortie mesurée et celle fournie par le modèle de référence pour la même entrée.

De ce fait, le contrôleur peut suivre l'évolution dynamique du bioprocédé à contrôler grâce aux prévisions du modèle de référence. Ici, contrairement aux MIAC, les mécanismes d'ajustement sont reliés aux entrées et sorties du système à contrôler, et utilisent les informations fournies par le modèle de référence. Ces informations permettent aux mécanismes d'ajustement de mettre à jour les paramètres du contrôleur.

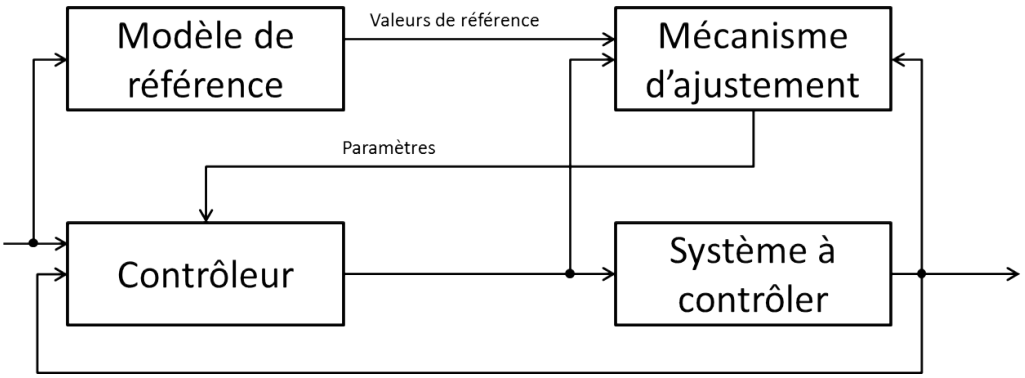

Figure III.2 – Fonctionnement d'un MRAC

III.2.3 Dual Control Theory

Une troisième approche de contrôle adaptatif est offerte par la « Dual Control Theory ». Issue des travaux de [Feldbaum 1961], cette approche est utilisée lorsque l'on doit contrôler un système inconnu. Dans ce cas, les actions de contrôle entreprises peuvent être divisées en deux catégories :

- Les contrôles proprement dits, donc l'objectif est de faire converger le système vers l'état souhaité ;
- Les sondes, permettant au contrôleur d'agir sur le système pour en observer les modifications et affiner son image du procédé à contrôler à partir de ces données.

Nous remarquons donc que, contrairement aux MIAC utilisant les observations des erreurs entre prédictions et mesures pour mettre à jour le modèle utilisé, le contrôleur réalise ici de véritables actions de contrôle, lui permettant d'acquérir les informations nécessaires à l'obtention d'une meilleure connaissance du système qu'il doit contrôler.

Ce type de contrôle s'éloigne donc des deux précédents, et se trouve particulièrement adapté lorsqu'un contrôle doit être réalisé dans des horizons de temps réduits et qu'il est nécessaire de converger rapidement vers une valeur même si ce n'est pas de manière optimale. L'utilité de cette approche apparaît également lorsque les paramètres du procédé étudié changent rapidement [Wittenmark 2002]. Cependant, l'utilisation d'actions de contrôle permettant d'obtenir des informations sur le système à contrôler peut globalement réduire la qualité du contrôle, entraînant des modifications au mieux non optimales sur le système : ces actions peuvent en effet modifier l'état du système à contrôler, le conduisant ainsi vers un état à partir duquel un résultat optimal ne peut plus être atteint. De plus, cette méthode est la plus complexe à mettre en œuvre et l'idée sous-jacente, bien que conceptuellement très intéressante, demande un grand travail d'instanciation en vue d'équilibrer les actions de contrôle et de recherche d'information.

III.2.4 Application à la biologie : la commande prédictive

Une fois appliqués au contrôle de procédés en général, ces trois mécanismes théoriques de contrôle adaptatif centrés sur l'utilisation de modèles sont regroupés sous le terme de Commande Prédictive (« Model Predictive Control » ou MPC en anglais) [Nikolaou 2001]. Cette approche permet de réaliser des actions de contrôle en temps réel guidées par des objectifs à plus long terme, tout en prévoyant et évitant d'éventuelles futures violations de contraintes.

L'intérêt principal de l'utilisation de modèles pour le contrôle de bioprocédés est sa capacité à gérer les interactions entre plusieurs variables de manière plus efficace que les régulateurs PID [Huang 2002]. En effet, les MPC sont capables de contrôler simultanément de multiples entrées et d'observer la différence entre les prévisions effectuées sur les sorties et les valeurs mesurées.

Le fonctionnement d'un MPC est relativement simple et se base sur l'utilisation d'un modèle afin de prédire l'évolution des variables de sortie selon l'état des entrées. Ainsi, le système se charge de minimiser l'erreur de la trajectoire prédite relativement à celle attendue sur un horizon de temps défini. Selon la nature du procédé à contrôler, les modèles utilisés peuvent indifféremment être linéaires ou non, et de ce fait, prendre différentes formes ; par exemple, celle d'un système d'équations différentielles ou d'un réseau de neurones. Ces modèles permettent d'estimer les valeurs futures prises par l'ensemble des variables. Ainsi, la connaissance des

objectifs à atteindre et des valeurs courantes et futures permet d'évaluer les modifications à entreprendre en vue de permettre au système de converger vers ces objectifs. L'algorithme général d'un MPC est illustré par la figure III.3.

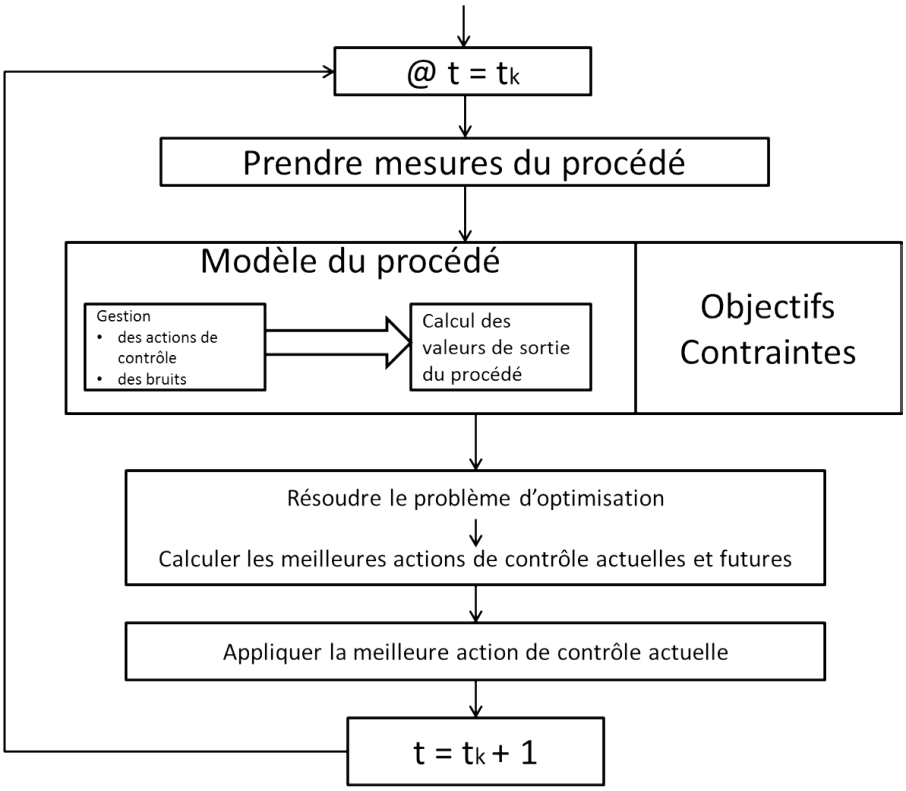

Figure III.3 – Fonctionnement d'un MPC

L'ajustement d'un tel mécanisme de contrôle s'appuie sur une variable nommée « pénalité de mouvement ». Afin d'assurer un contrôle robuste, les contrôles appliqués sur le bioprocédé par le contrôleur entraînent une pénalité. Plus précisément, chacun des éléments sur lesquels le contrôleur peut agir est lié à une certaine valeur de pénalité qui représente l'impact d'une modification de cet élément. Cette pénalité de mouvement permet de limiter ainsi les modifications effectuées sur le bioprocédé et de pondérer les différents choix possibles, ce qui conduit à des comportements de contrôle distincts. Une pénalité élevée aura tendance à produire un contrôleur stable, effectuant ses actions lentement, tandis qu'une pénalité faible accélère la réponse du contrôleur tout en limitant sa marge de stabilité.

Tableau III.4 – Récapitulatif des caractéristiques du contrôle adaptatif

Nom du contrôle / Critère de comparaison	Contrôle adaptatif
Adaptabilité	++
Généricité	- (dépendant de la détermination du modèle)
Position	Globale : Supervision
Remarques	La généricité de ces approches augmente en même temps que la complexité de leur mise en œuvre

III.2.5 Avantages et limites

Ces approches de contrôle adaptatives proposent des solutions adéquates au contrôle de systèmes complexes et sont dotées de capacités d'adaptation dont les régulateurs PID sont dépourvus. Malheureusement, ces avantages issus de l'utilisation d'un ou plusieurs modèles forment également leur principal défaut. En effet, la modélisation de bioprocédés est une étape mathématiquement très complexe. A cette complexité s'ajoutent les éventuelles contraintes sur les types de modèles à utiliser avec l'approche de contrôle adoptée, contraignant d'autant plus la phase de modélisation et, de ce fait, limitant l'utilisation de ce type de contrôle. Globalement, le travail nécessaire afin de les appliquer à des bioprocédés de grande complexité ou peu étudiés se révèle démesuré.

Cette limite reste malgré tout à nuancer car elle dépend essentiellement de la modélisation de bioprocédés et non directement du contrôle de ceux-ci.

Enfin, les approches permettant de s'affranchir de la création préalable de modèles, telles que la Dual Control Theory, compensent cet avantage par une phase de modélisation automatique souvent très complexe, et donc, peu applicable à des problèmes réels tels que le contrôle de bioprocédés.

La table III.4 résume les caractéristiques principales des techniques de contrôle adaptatif, notamment concernant leur généricité, adaptabilité et complexité.

III.3 Le contrôle intelligent

Le dernier type de contrôle est dit intelligent. Il se focalise sur l'utilisation de certaines techniques issues de l'intelligence artificielle afin d'offrir des mécanismes d'adaptation efficaces

et de gérer les non-linéarités de manière satisfaisante. En ce sens, cette catégorie de contrôleurs peut être considérée comme un sous-ensemble des mécanismes de contrôle adaptatif. Une grande diversité de techniques sont mises en œuvre pour atteindre ces objectifs, les plus célèbres et également les plus utilisées font appel aux réseaux de neurones [Miller 1995], aux probabilités bayesiennes, à la logique floue [Lee 1990] et aux systèmes experts [Stengel 2002].

III.3.1 Réseaux de neurones

Les réseaux de neurones sont des modèles de calcul particulièrement utilisés dans les domaines de la classification, de la robotique et, plus généralement, dans tout domaine dont la complexité rend, en général, impossible la création d'une fonction de transformation des valeurs d'entrées en valeurs de sorties.

Leur application au contrôle des bioprocédés permet principalement d'inférer des valeurs non mesurables, mais également de contrôler certains aspects de ces bioprocédés, caractéristique particulièrement utile dans les cas où la complexité ou la non-linéarité du bioprocédé s'opposent à l'utilisation des méthodes de contrôle conventionnelles telles que l'utilisation de régulateurs PID.

Un réseau de neurones est un système de calcul dont le traitement est inspiré de celui effectué par les neurones biologiques. Les unités de ce réseau, appelées neurones, sont liées entre elles de manière pondérée et produisent des sorties dépendant des entrées reçues à l'aide de leurs propres fonctions de transfert, ou fonctions seuils locales. Le poids appliqué à un lien représente la force de celui-ci, chacun des liens entre neurones, illustrés sur la figure III.4, possède son poids propre. Biologiquement parlant, ce poids représente une sorte de force synaptique du neurone.

Les neurones composant le réseau sont organisés en plusieurs couches, parmi lesquelles nous distinguons habituellement les couches entrée, sortie et intermédiaires ; ces dernières sont également appelées couches cachées. Lorsqu'ils sont considérés individuellement, ces neurones ont une capacité de calcul limitée, mais du réseau formé par leurs transmissions d'informations émerge une fonction capable de traiter des problèmes complexes.

De tels systèmes acquièrent leur comportement non pas grâce à une série de règles, mais en étant confrontés à des données issues du problème qu'ils auront à traiter. Cette phase est nommée apprentissage et permet au réseau de neurones d'ajuster la valeur de ses poids de manière à obtenir des résultats similaires à ceux lui étant présentés. Différentes méthodes d'apprentissage peuvent être utilisées, telles que les apprentissages supervisés, non supervisés et par

renforcement. L'ajustement des valeurs des poids est calculé en utilisant des méthodes de type « descente en gradient » et, principalement, des algorithmes de rétro-propagation qui renvoient l'erreur par les différents neurones ayant mené à celle-ci. Cependant, cette phrase d'apprentissage est critique à cause du risque de sur-apprentissage. En effet, lorsqu'un réseau de neurones est entraîné à partir d'un trop grand nombre de données, son fonctionnement peut alors se « figer » dans un type de comportement appris, l'amputant ainsi de son adaptabilité. À l'inverse, un mauvais choix des données d'apprentissage peut également souligner des phénomènes liés au bruit des mesures et diriger le réseau de neurones vers l'adoption d'un comportement spécifique et erroné pour une gamme donnée de valeurs.

Une fois l'apprentissage réalisé, le réseau de neurones est confronté à un jeu de données test différent des données d'apprentissage. Ceci permet sa validation dès lors que l'erreur sur les résultats obtenus devient inférieure à un seuil défini.

Au-delà de ce mécanisme global commun, différents types de réseaux de neurones peuvent être établis selon de nombreux critères tels que, par exemple, leur nombre de couches, leurs possibles rebouclages et les fonctions d'agrégation permettant de combiner les valeurs d'entrées reçues par les neurones. Le réseau de neurones le plus simple est le perceptron qui est un réseau formé d'une seule couche, sans cycle et dont les neurones utilisent des fonctions seuils. Généralement, ce type de réseau possède une sortie unique, reliée à toutes les entrées [Rosenblatt 1958].

Ce modèle a été amélioré pour former le perceptron multi-couche dont la particularité est d'utiliser des fonctions d'activation non linéaires et d'être un approximateur universel [Hornik 1993]. La figure III.4 illustre son fonctionnement.

Ces réseaux de neurones sont la méthode informatique centrée sur l'utilisation d'une boîte noire la plus répandue dans le domaine du contrôle. Ils ont connu de très nombreuses améliorations afin de remplir cette tâche, en particulier leur combinaison avec des équations permettant ainsi d'obtenir des « gray-box », diverses techniques d'apprentissage et l'utilisation de « recurrent neural networks » [Williams 1989]. De plus, de très nombreux systèmes de contrôle hybrides utilisent les réseaux de neurones à des fins diverses, allant de la modélisation de bioprocédés à la reconnaissance des formes des courbes représentant l'évolution des variables de bioprocédés. Ainsi, l'utilisation d'un réseau de neurones dans le processus général de contrôle de bioprocédés est chose courante même si ce réseau n'est pas utilisé en tant que contrôleur proprement dit.

Cependant, malgré ses avantages, l'application de réseaux de neurones au contrôle de bioprocédés entraîne certaines contraintes, notamment dues à l'importante quantité de données

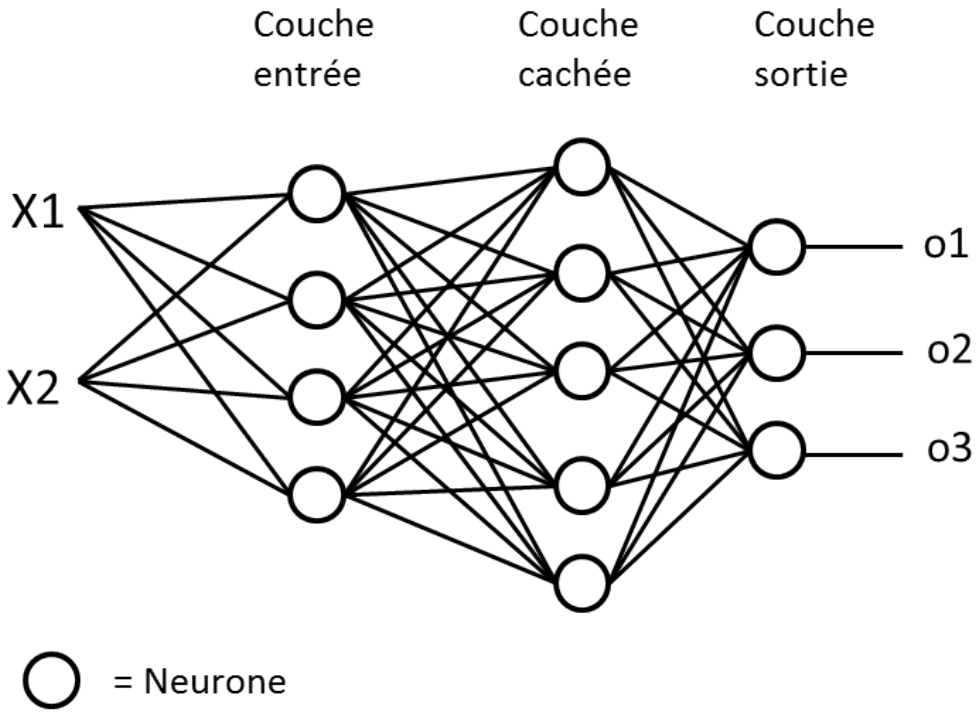

Figure III.4 – Structure d'un réseau de neurones perceptron multi-couche

nécessaires à leur apprentissage, données limitées dans le cadre du contrôle de bioprocédés, mais aussi à la plage de valeurs réduite sur laquelle ces réseaux s'avèrent capables d'effectuer leurs contrôles. En effet, un réseau de neurones étant dépendant de son apprentissage, il est sensible aux situations qu'il n'a jamais rencontrées et, de ce fait, si une variable sort de l'intervalle des valeurs attendues (car rencontrées lors de son apprentissage), il est possible que le réseau ne soit pas en mesure de faire converger à nouveau le système vers l'état souhaité.

III.3.2 Contrôleurs bayésiens

Les contrôleurs bayésiens [Colosimo 2006] reposent sur les probabilités bayésiennes établissant que, pour des événements dépendants, la probabilité qu'un événement se produise à l'avenir peut être déduite des occurrences précédentes de cet événement.

Les plus célèbres d'entre eux, bien que n'ayant pas été découverts selon une approche bayésienne, et découlant des théories du contrôle moderne, sont les filtres de Kalman et les filtres de Kalman étendus [Lee 1994], respectivement appliqués à la résolution de problèmes linéaires et non-linéaires. Ces filtres, également appelés « équations linéaires quadratiques », permettent

d'estimer à partir de mesures bruitées l'état dans lequel le système se trouve. Le principe de ce filtrage est de distinguer deux phases. La première phase, dédiée à la « prévision », permet de produire une estimation de l'état courant à partir de l'estimation effectuée de l'état précédent. La seconde phase, concernant la « mise à jour », utilise les observations afin d'affiner l'estimation de l'état courant prédit.

De nombreuses modifications et améliorations reposent sur ces filtres de Kalman, en particulier les ensembles de filtres de Kalman [Evensen 2003] et les filtres de Kalman non parfumés [Julier 2005] qui permettent une gestion efficace des cas fortement non linéaires et suppriment le besoin de calcul des jacobiens.

Les filtres de Kalman ne sont pas une méthode de contrôle à part entière, mais permettent de jouer le rôle d'observateur d'état, filtrant les données issues des capteurs et conduisant ainsi à la création d'une commande adaptée à la situation courante. Ils agissent en tant que composant d'un contrôleur.

En ce sens, ces filtres apparaissent au niveau du traitement des données observées ou la création de modèle, mais leurs approches décomposant le processus en deux phases « prévision » et « observation » est intéressante car offrant une certaine modélisation indirecte de l'évolution des valeurs du système à contrôler. Les filtres de Kalman, en particulier leurs versions étendues, sont une approche plus particulièrement utilisée en maitrise statistique des procédés.

Il est à noter que les approches bayésiennes ne se limitent pas à l'utilisation de ces filtres et contiennent également des techniques issues de la famille des filtrages particulaires [Carpenter 2002].

III.3.3 Systèmes experts

Issus des travaux en intelligence artificielle, les systèmes experts apparaissent également à de multiples niveaux dans le contrôle de bioprocédés. Leur principe général de fonctionnement se base sur l'utilisation de deux éléments : une base de connaissance et un moteur d'inférence.

La base de connaissance, composée de faits et de règles, représente l'ensemble du savoir du système. Le moteur d'inférence pour sa part, indique au système expert la manière d'utiliser la base de connaissance afin de répondre aux demandes de l'utilisateur, en manipulant et combinant les règles et les faits qu'elle contient.

Ainsi, les systèmes experts peuvent se situer au niveau de la conduite de bioprocédés en jouant le rôle de l'opérateur afin de décider des objectifs à atteindre et de l'approche à adopter selon l'évolution des variables observées. Cette utilisation implique donc une observation du

procédé dans son ensemble. De manière similaire, un système expert peut également se situer au niveau d'une variable isolée, permettant le contrôle de celle-ci comme le ferait un régulateur PID [Konstantinov 1993].

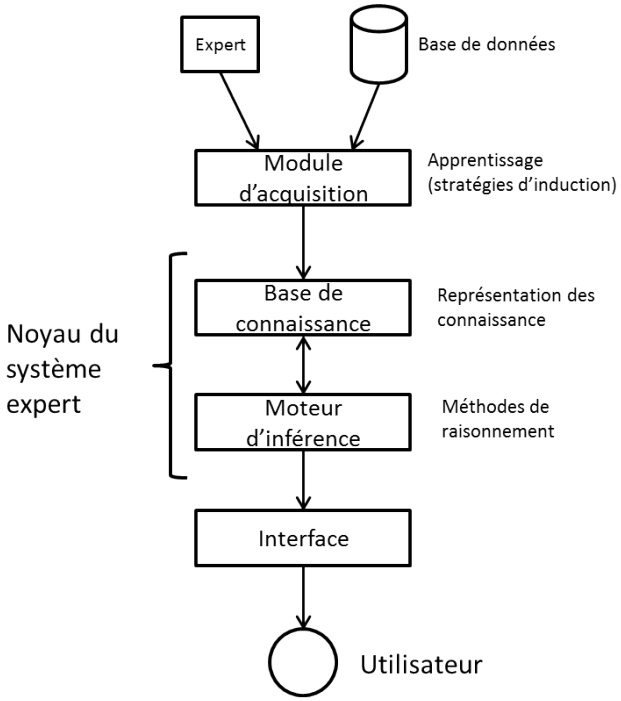

Figure III.5 – Fonctionnement d'un système expert

De plus, les systèmes experts peuvent jouer un rôle de supervision important en communiquant avec l'humain contrôlant le bioprocédé et en l'avertissant en cas de pannes, par exemple, ou de tout autre comportement observé qu'il considère comme anormal. Ce type d'interaction est représenté par la figure III.5, illustrant les différentes étapes permettant le passage de l'information d'une base de connaissance jusqu'à l'utilisateur. La base de connaissance est composée de données issues d'experts, acquises à l'aide d'un module spécifique. Ces données sont ensuite traitées à l'aide d'un moteur d'inférence permettant d'en déduire de nouvelles règles. Une fois que le système expert détecte un évènement quelconque sur le bioprocédé, il utilise son moteur d'inférence afin d'en déduire les conséquences et si besoin, alerte l'utilisateur d'une action de la nécessité d'entreprendre une action spécifique.

Cependant, une telle supervision nécessite un apprentissage spécifique sur un procédé précis et, de ce fait, cette approche présente un certain manque de généricité et s'adapte mal aux systèmes fortement dynamiques et non linéaires. Ce phénomène est renforcé par un manque d'adaptabilité ce qui requiert une modification de la base de connaissance lorsque le système

est confronté à un événement nouveau.

III.3.4 Logique floue

La logique floue est une approche issue de l'intelligence artificielle, développée par [Zadeh 2002], et centrée sur la théorie des ensembles flous. L'apport de cette logique, par rapport à la logique classique, est l'apparition d'états intermédiaires entre le Vrai et le Faux. Cette caractéristique apparaît comme un élément essentiel permettant l'élaboration de contrôleurs flous (Fuzzy Logic Controller, FLC), dont le raisonnement s'adapte mieux à l'incertitude, et se rapproche des comportements humains de prise de décisions. De ce fait, l'utilisation conjointe de mécanismes de logique floue et de système experts permet la création de systèmes de supervision et de contrôle dont le comportement s'adapte à la dynamique des bioprocédés.

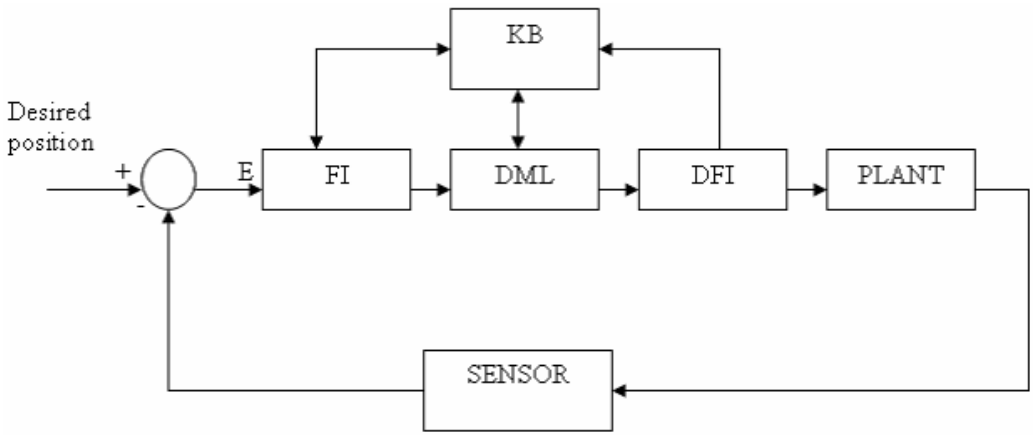

Figure III.6 – Fonctionnement général d'un FLC

Le fonctionnement général d'un FLC, illustré par la figure III.6, est composé des éléments suivants :

- Une « interface floue » (FI) qui convertit le signal des capteurs en signal pertinent pour être intégré à un raisonnement centré sur la logique floue.
- Une « base de connaissances » (KB), similaire à celle utilisée par les systèmes experts, contenant les règles floues et les faits à utiliser, ainsi que les objectifs à atteindre.
- Une « unité de prise de décision logique » (DML) qui utilise les connaissances de la KB et les données de la FI pour établir une décision à l'aide d'un raisonnement flou.
- Une « interface de transformation » qui convertit les décisions de la DML en contrôles applicables sur le procédé, c'est-à-dire en valeur réelles.

Au-delà des liens entretenus entre la logique floue et les systèmes experts, la logique floue peut également être couplée à d'autres types de contrôleurs, en particulier aux régulateurs PID [Visioli 2001]. Ainsi, une part d'adaptation et une prise de décision plus humaine sont ajoutées aux régulateurs, au prix d'une implémentation plus complexe. La logique floue permet alors, à l'image des méthodes de Ziegler-Nichols et Cohen-Coon, d'ajuster dynamiquement les différentes variables apparaissant dans les équations des régulateurs PID, permettant ainsi au régulateur de s'adapter plus rapidement aux changements du bioprocédé.

III.3.5 Avantages et limites

Le contrôle intelligent regroupe donc une grande variété d'approches distinctes dont les avantages respectifs sont nombreux. Outre le caractère adaptatif dont elles font preuve, ces approches présentent également une caractéristique notable : la facilité à les combiner entre elles, ou avec des méthodes de contrôle plus statiques. De ce fait, leur présence dans les systèmes de contrôle de bioprocédés peut se faire à de multiples niveaux, que ce soit pour réguler une variable isolée ou pour aider l'utilisateur dans la conduite globale du bioprocédé. C'est pourquoi ces approches se caractérisent par une généricité supérieure à celle des techniques classiques de contrôle adaptatif.

Bien que possédant de nombreuses qualités, en particulier dans leur adaptabilité, ces méthodes intelligentes regroupent principalement des approches dites « boîte noire » et leur application industrielle aux contrôles de procédés reste frileuse. En effet, aujourd'hui encore, peu de confiance est accordée au caractère prédictif de ces approches, en particulier celui des réseaux de neurones. De ce fait, ils apparaissent souvent en complément de systèmes mathématiques plus classiques, offrant au final des contrôleurs de type « boîte grise ». Ces différences sont illustrées par la figure III.7.

Plus généralement, ces approches de contrôle intelligent dépendent fortement de la phase d'apprentissage ou de leur base de connaissances, et se retrouvent ainsi limitées en cas de comportements imprévus. De plus, le travail de modélisation de tels systèmes reste conséquent et établir une base de connaissances ou un modèle fidèle d'un bioprocédé est une phase difficile, complexifiant encore l'adoption de ce type d'approches. De même, le travail d'instanciation et d'ajustement réalisé sur un problème est en grande partie perdu si l'on souhaite s'intéresser à un procédé différent.

Malgré tout, le point essentiel de ces approches intelligentes réside dans le nombre de contextes différents dans lesquels elles sont applicables, et dans leur capacité d'améliorer l'adap-

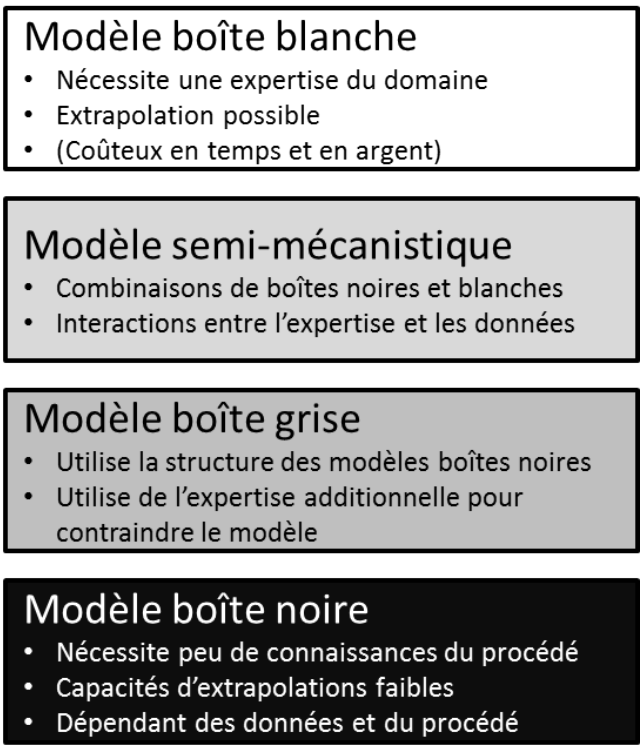

Figure III.7 – Classification des modèles

tation des systèmes auxquels elles s'ajoutent, qu'ils s'en servent comme base principale de raisonnement ou d'outil afin d'améliorer leur compréhension des données reçues. Ces différents points sont résumés dans le tableau III.5.

III.4 Les agents et le contrôle

Certains travaux ont permis d'explorer l'utilisation d'un système multi-agent pour contrôler un procédé. Parmi les travaux existants, on peut citer [Guo 2006] qui traite du contrôle de la pression de gaz recyclés, [Davidsson 2002a] qui présente un exemple du contrôle d'un réseau de chauffage et [Cockburn 1992] qui s'attache à gérer la distribution d'électricité dans un réseau. Dans chacun de ces cas, un SMA a été développé pour mener à bien ces contrôles. Ces SMA diffèrent tant par les agents utilisés que par leur organisation et les différentes architectures multi-agents utilisées sont décrites dans les paragraphes qui suivent.

Si l'on se ramène au problème du contrôle qui nous préoccupe dans ce mémoire, utiliser un SMA pour contrôler spécifiquement un bioprocédé est une idée récente et très peu traitée

Tableau III.5 – Récapitulatif des caractéristiques du contrôle intelligent

Nom du contrôle / Critère de comparaison	Contrôle intelligent
Adaptabilité	++
Généricité	+ (sous réserve de données d'apprentissage disponibles)
Position	Aussi bien au niveau local en complément d'autres approches, que global pour la supervision
Remarques	Aspect boîte noire limitant

dans la littérature. En effet, l'application d'un SMA à cette problématique est essentiellement envisagée de manière indirecte.

C'est le cas de [Davidsson 2002b], qui justifie la pertinence des SMA dans ce cadre précis de par la proximité du problème avec le contrôle de procédés au sens général (non biologiques). L'un des arguments présenté réside dans l'adéquation entre le problème du contrôle de bioprocédés et les critères de [Parunak 2000] définissant les situations dans lesquelles l'utilisation d'un SMA est pertinente.

En outre, les possibilités méthodologiques offertes par le niveau d'abstraction des SMA constituent également un avantage non négligeable. La décomposition d'un problème aussi complexe que le contrôle d'un bioprocédé permet une meilleure compréhension de son fonctionnement ainsi qu'un traitement simplifié par la résolution de sous-problèmes plus accessibles.

L'un des travaux les plus aboutis concernant le contrôle de bioprocédés est réalisé par [Gao 2010]. Ce dernier présente un SMA permettant de modéliser un bioprocédé et d'effectuer des tâches de surveillance simplifiant la prise de décision d'un utilisateur. Le détail de cette architecture est commenté dans la section III.4.4.

III.4.1 Bâtiments intelligents

Un SMA destiné au contrôle d'immeubles est présenté par [Davidsson 2000]. Son objectif vise à permettre une réduction de la consommation électrique tout en améliorant le confort des personnes travaillant à l'intérieur du bâtiment.

Pour ce faire, ce SMA se base sur la communication des différentes entités existant dans le bâtiment, telles que les capteurs, les interrupteurs, le chauffage ou la ventilation, en vue d'adapter leurs comportements. Le système réalisé comprend quatre types d'agents distincts.

- Un agent est créé pour chacune des pièces du bâtiment. Son objectif est de diminuer au maximum la consommation énergétique de la pièce tout en maintenant la satisfaction des personnes s'y trouvant.
- Un agent est créé pour chacun des utilisateurs, afin de stocker ses préférences (température idéale par exemple) et d'agir en son nom lors des communications entre agents.
- Un agent est créé pour chaque paramètre environnemental d'une pièce particulière. Ces agents ont accès aux capteurs et actuateurs correspondant à leurs paramètres associés. Par exemple, un agent température correspond à un capteur mesurant la température et à l'actuateur correspondant au contrôle du radiateur de cette pièce.
- Le dernier type d'agents correspond à un système de badges, portés par chacune des personnes présentes dans le bâtiment et permettant de les situer par rapport aux différentes pièces.

Ainsi, les communications entre agents permettent aux agents des pièces d'assurer leur objectif énergétique, tout en tenant compte des conditions environnementales et des attentes des utilisateurs. L'ensemble des comportements des agents assure donc, comme [Davidsson 2000] le présente lors d'une application pratique du système, un équilibre entre satisfaction des utilisateurs et économie d'énergie.

L'intérêt majeur de cette approche réside dans sa généricité. Ceci permet son application à un très grand nombre de bâtiments aux caractéristiques différentes tout en offrant un ajustement des paramètres plus précis que ne le permettent les approches usuelles.

Cependant, malgré cette généricité, le problème adressé demeure statique. Les agents n'ont de ce fait aucune nécessité d'apprentissage, toutes les informations sur lesquelles leur raisonnement se base étant déterminées a priori. Une telle architecture n'est donc pas applicable au contrôle de bioprocédés car rien ne permet aux agents de s'adapter à une nouvelle situation.

III.4.2 Distribution d'électricité

[Cockburn 1992] présente un SMA, nommé CIDIM (Cooperating Intelligent System for Distribution System Management), destiné à contrôler la distribution d'électricité en se basant sur la plateforme ARCHON [Jennings 1991].

Le fonctionnement de ce système repose sur la coopération entre dix agents aux rôles distincts :

- Agent de télémétrie : ce premier agent est chargé de la réception, de l'uniformisation des données de télémétrie et de leur transmission aux autres agents.

- Agent d'information : cet agent permet l'accès à une base de données décrivant toutes les informations disponibles sur le réseau à contrôler comme, par exemple, la connectivité des pièces le composant.
- Agent d'observation météorologique : cet agent gère les données concernant les impacts de foudre et les situe sur le réseau électrique.
- Agent sécurité : cet agent est responsable de la prévision d'éventuelles surcharges causées par des dégâts sur le réseau.
- Agent de diagnostic haut voltage : cet agent utilise les données de l'agent de télémétrie afin de diagnostiquer les lieux, types et temps des erreurs apparaissant sur le réseau. De plus, cet agent permet de détecter les zones non alimentées.
- Agent de diagnostic bas voltage : cet agent possède le même rôle que l'agent de diagnostic haut voltage, mais sur un type de réseau différent. Il se base sur les données issues des appels téléphoniques et sur celles provenant de l'agent d'observation météorologique.
- Agent de planning d'intervention : ce type d'agent prévoit les interventions des techniciens sur certaines parties du réseau. Il s'assure que celles-ci sont bien isolées au moment de leurs modifications.
- Agent de vérification d'intervention : cet agent vérifie la possibilité d'intervention sur une section particulière du réseau.
- Agent de supervision : cet agent reçoit les rapports d'erreur des deux agents de diagnostic, ainsi que les plannings d'intervention. Il réalise une mise en relation de ces données afin de déterminer, par exemple, si une opération doit être annulée suite à un accident sur le réseau.
- Agent de conseil : ce dernier agent est en fait l'interface du SMA via laquelle un opérateur va l'utiliser et accéder à ses données.

Les communications entre ces différents agents sont assurées par la plateforme ARCHON [Jennings 1991], qui encapsule les agents et permet la coopération entre différentes entités distinctes au sein d'un même système.

L'utilisation d'un tel SMA permet une amélioration notable de la robustesse du système car ce dernier devient capable de fournir des résultats partiels ou corrigés lors de la détection d'erreurs sur les agents le composant.

De plus, ce SMA repose sur un mécanisme de coopération entre agents afin de résoudre des situations pour lesquelles un agent unique est insuffisant, comme c'est le cas par exemple lorsqu'il s'agit de fournir une réponse à une requête complexe d'un utilisateur.

Le SMA présenté est donc composé d'un nombre fixe d'agents très spécialisés. Il n'existe

qu'une seule instance de chacun des types d'agents et les relations entre eux sont statiques tout au long du fonctionnement du système.

Cet aspect statique se présente comme un frein à la généricité recherchée afin de résoudre le problème du contrôle de bioprocédés. En effet, le SMA réalisé par [Cockburn 1992] se base sur une décomposition du problème de gestion d'un réseau électrique provenant des entités physiques existant dans ce domaine d'application précis. Les interactions entre les agents apportent des résultats de meilleure qualité, mais ceux-ci restent principalement dépendants du comportement des agents considérés de manière isolée. Or, de par la spécification importante de ces agents, il n'est pas possible d'étendre ce système afin d'en extraire un fonctionnement générique permettant son application au problème précis du contrôle de bioprocédés.

III.4.3 Contrôle de gaz recyclés

Les travaux de [Guo 2006] présentent une méthode de contrôle dite hybride, car utilisant des unités de contrôle décentralisées (les agents) dans une organisation hiérarchique. Cette organisation émerge de par les interactions entre agents permettant de leur assigner un rôle spécifique conduisant à une structure hiérarchique dynamique sur deux niveaux, évoluant selon l'état du système.

Chaque agent du SMA possède la même structure, ainsi que le même ensemble de fonctions de base. A partir de ce modèle d'agents génériques, différentes instanciations sont réalisées pour gérer en particulier des problèmes d'un domaine précis. Par exemple, un type d'agent peut être instancié pour représenter le comportement à appliquer en cas de détection d'une anomalie spécifique. La création des agents du SMA nécessite donc d'être en mesure de décomposer le problème global du contrôle en une série de sous-problèmes couvrant l'intégralité des anomalies éventuelles que le système peut rencontrer.

Ces agents collaborent ensuite selon une extension du protocole de réseau contractuel [Hsieh 2002], leur permettant de résoudre des problèmes qu'ils ne peuvent traiter individuellement. Le fonctionnement de ce protocole est le suivant : chaque agent possède un rôle, gestionnaire ou contractant. L'agent devant exécuter une tâche (le gestionnaire) décompose cette tâche en plusieurs sous-tâches. Le gestionnaire annonce ensuite chaque sous-tâche aux contractants. Ceux-ci évaluent alors l'annonce et s'ils en ont les capacités, répondent au gestionnaire qu'ils sont en mesure de l'effectuer. Le gestionnaire et le contractant échangent alors les informations nécessaires à l'exécution de cette tâche.

Les agents agissant comme gestionnaires forment donc le haut de la hiérarchie du SMA, tandis que les contractants en constituent la base.

A l'inverse des exemples précédents, cette approche propose l'utilisation d'agents dérivant d'une interface commune et possédant le même comportement. De plus, l'organisation hybride mise en place offre une dynamique permettant au SMA de s'adapter aux changements observés sur le système à contrôler. Ces points constituent des avantages non négligeables et permettent d'envisager l'utilisation d'une structure similaire afin de contrôler un bioprocédé.

Malheureusement, deux points viennent freiner cette application. Tout d'abord, cette approche nécessite une étape de décomposition du problème global en sous-problèmes, celle-ci n'étant pas effectuée directement par les agents. Cette étape impose de connaître l'ensemble des perturbations auxquelles le système à contrôler sera éventuellement soumis, ainsi que les actions à entreprendre afin d'y répondre. A cette limite s'ajoute l'absence de capacités d'apprentissage permettant aux agents de découvrir eux-mêmes comment réagir en cas de situations non spécifiées lors de leurs créations. Le problème du contrôle de bioprocédés empêchant de fournir aux agents cette connaissance complète, une telle approche ne peut donc pas être appliquée sans de nombreuses modifications.

III.4.4 Contrôle de bioprocédés à l'aide d'un SMA

L'exemple d'application d'un SMA au contrôle de bioprocédés le plus richement développé est issu de [Gao 2010]. Le système proposé dans cet article se base sur le constat qu'utiliser un SMA peut permettre d'améliorer deux aspects distincts en rapport avec le contrôle d'un bioprocédé.

Le premier d'entre eux réside dans l'étape de conception du bioprocédé lui-même. Cette étape est particulièrement complexe de par la modélisation du bioprocédé qu'elle impose. En effet, cette modélisation correspond à la séparation des différents composants en interaction à l'intérieur du bioprocédé, couplée à la modélisation des interactions elles-mêmes. Cette étape fastidieuse présente des caractéristiques de distribution et d'interaction adéquates avec l'utilisation d'un SMA, et pourrait donc bénéficier de son application.

Le second aspect concerné par l'utilisation d'un SMA apparaît lors du déroulement du bioprocédé. Durant cette étape, la détection d'erreurs est primordiale et il reste nécessaire de fournir des conseils à l'utilisateur afin de le guider quant aux actions à entreprendre vis-à-vis des évènements se produisant dans le bioprocédé.

Afin d'atteindre ces deux objectifs, aider à la modélisation du bioprocédé et fournir une aide à l'utilisateur, une architecture unique de SMA a été conçue. Celle-ci, présentée dans la figure III.8, est composée des quatre types d'agents suivants :

- Agent « First Principle » (FP) : ce type d'agent peut accéder aux données expérimentales ainsi qu'aux modèles physiques développés. Son rôle est de choisir le modèle de la complexité correspondant aux attentes de l'utilisateur au sein d'une bibliothèque de modèles.
- Agent « Artificial Intelligence » (AI) : ce type d'agent repose sur l'utilisation de modèles de type « boîtes noires » afin d'effectuer des simulations et des prévisions sur les performances du bioprocédé. Un agent AI peut également être utilisé pour mettre à jour les modèles utilisés par le SMA, ainsi que la description du bioprocédé.
- Agent « Unit Operation » (UO) : ce type d'agent permet de représenter des opérations spécifiques sur le bioprocédé. Ces opérations sont déterminées par une division du problème global en plusieurs sous-problèmes. Ainsi, de telles opérations peuvent être, par exemple, les étapes de fermentation, de centrifugation ou de chromatographie. Le rôle de cette catégorie d'agent est double. D'une part, il permet de lier les résultats provenant des agents AI et FP, et offre ainsi une image plus précise de ce qui se passe dans le bioprocédé, et d'autre part, il peut communiquer avec d'autres agents UO afin d'établir une évaluation des paramètres du bioprocédé tenant compte des états des différentes unités d'opération définies.
- Agent « Coordination » (C) : ce dernier type d'agent se situe au sommet de la hiérarchie du SMA. Son rôle est de diviser une tâche en plusieurs sous-tâches qui seront transmises aux agents UO correspondants. Cet agent est unique dans l'architecture du SMA : il permet d'aider l'utilisateur dans sa prise de décision et de définir la politique opératoire à appliquer pour optimiser la production.

L'organisation hiérarchique du SMA permet de limiter le nombre d'agents devant traiter l'importante quantité de données provenant des modèles mathématiques ou des résultats de simulations effectuées par les modèles boîtes noires. Malgré sa structure hiérarchique le SMA conserve une certaine modularité dans le sens où les agents UO (et par extension, FP et AI) peuvent être ajoutés ou supprimés selon le problème à traiter.

Le système ainsi conçu est appliqué sur un problème typique de production de protéines intracellulaires : la production d'alcool déshydrogénase par une culture de Saccharomyces Cerevisiae ou plus communément appelée levure de bière.

Une structure de SMA a été construite en fonction de ce problème précis, en particulier grâce

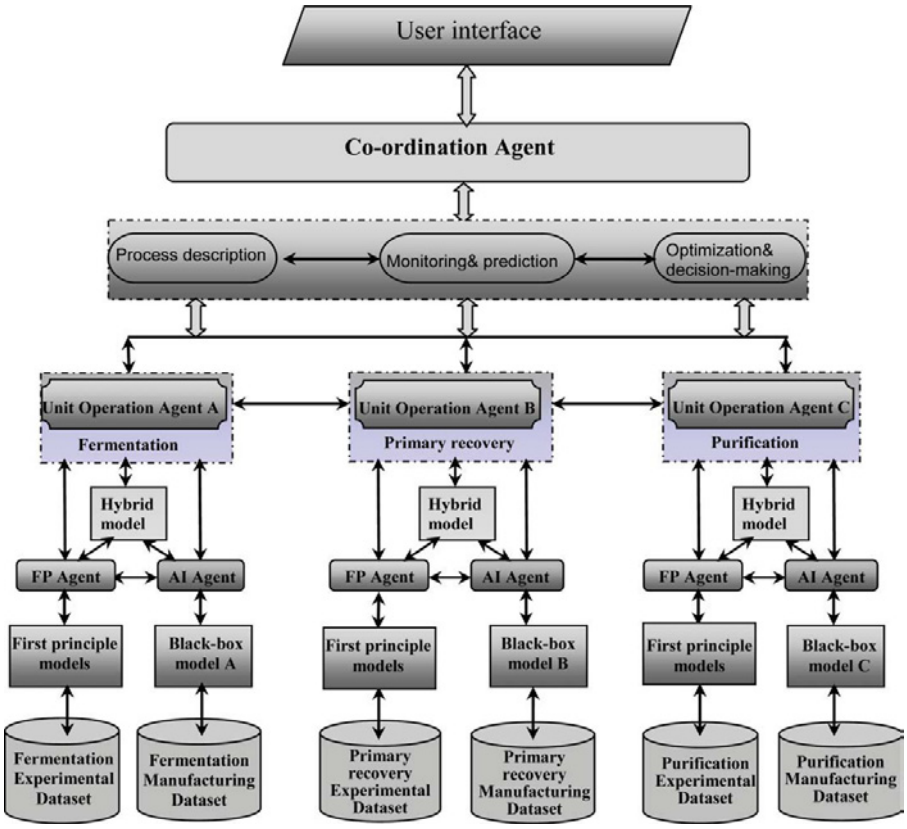

Figure III.8 – Architecture du SMA de contrôle de bioprocédés proposée par [Gao 2010]

à la définition de quatre agents UO, dédiés respectivement à la fermentation, à la centrifugation, à l'homogénéisation et à la chromatographie. Cette division est issue des connaissances biologiques des concepteurs du bioprocédé ainsi que de l'ensemble des modèles et des outils disponibles afin de permettre aux agents de pouvoir appliquer le comportement attendu.

Le SMA ainsi créé permet d'identifier les conditions dans lesquelles les agents UO permettent d'obtenir les meilleurs résultats selon une fonction objectif tenant compte de la productivité souhaitée. Ainsi, les différents agents UO se chargent d'appliquer un algorithme d'optimisation sur les modèles auxquels ils ont accès en tenant compte d'une plage de paramètres fournie par l'agent Coordination.

Une fois cette optimisation réalisée, le SMA est utilisé conjointement à une base hiérarchique de règles afin de surveiller le fonctionnement du bioprocédé, et d'alerter l'utilisateur en cas de divergence entre les prévisions et les mesures réalisées. Ces alertes s'accompagnent d'une suggestion d'action à réaliser pour corriger la divergence du bioprocédé. Deux exemples de tests ont permis de valider la pertinence de l'action suggérée par le SMA.

L'approche exposée dans ces travaux représente l'unique application détaillée de l'application d'un SMA au contrôle de bioprocédés, et se base sur un SMA flexible et capable d'effectuer un grand nombre d'actions capables d'aider l'utilisateur, tant dans la conception que lors du contrôle du bioprocédé.

Malheureusement, l'application d'un tel SMA demande de très nombreuses ressources, en particulier concernant les modèles utilisables par les agents. Il est nécessaire de disposer d'une importante quantité d'informations et ce, sous plusieurs formes, comprenant à la fois modèles mathématiques et boîtes noires. Ainsi, la qualité des résultats obtenus dépend fortement de cette base de connaissances initiale.

De surcroît, le contrôle d'un tel SMA est centralisé dans l'agent Coordination, la stabilité du SMA étant donc fortement réduite en cas de défaillance de ce dernier.

Cette approche reste la plus aboutie à ce jour et, bien que relativement flexible, elle manque de capacités d'adaptation et d'apprentissage la rendant générique afin d'éviter le recours à une collection de modèles difficilement accessibles. Ces limites sont donc un obstacle à son application dans le cadre de notre problématique.

III.4.5 Avantages et limites

Les problèmes précédemment décrits présentaient certains des prérequis (modularité, distribution, complexité, dynamicité) à l'utilisation d'un système multi-agent. Les solutions proposées marquent, en outre, un précédent quant à l'application réussie des systèmes multi-agents à des problèmes de contrôle de procédés en général, et de bioprocédés en particulier. Une fois comparées aux approches classiques de contrôle présentées au début de ce chapitre, les approches centrées sur l'utilisation de SMA présentent des caractéristiques d'adaptation améliorées, tout en évitant l'aspect boîte noire décrié. De plus, elles possèdent de multiples applications à la problématique du contrôle, allant des étapes de conception jusqu'aux conseils fournis à l'utilisateur en passant par l'ajustement automatique de paramètres. Ainsi, l'utilisation de SMA tend à limiter les défauts des approches issues de l'intelligence artificielle, tout en renforçant leurs aspects positifs.

Une limite essentielle est cependant commune aux différentes approches présentées : leurs capacités d'apprentissage limitées et, par extension, l'importante quantité de données nécessaires afin d'initialiser les agents. Que ces dernières prennent la forme de différents modèles du système à contrôler ou d'un ensemble de règles déterminant des comportements précis, cette

connaissance nécessaire aux SMA décrits n'est pas entièrement accessible dans le problème traité au cours de cette thèse. L'utilisation d'une grande quantité de données implique également qu'une grande partie des agents, présentés dans ces travaux, sont hautement cognitifs et possèdent des mécanismes complexes de raisonnement sur leurs connaissances. Cet aspect entraîne un besoin en calcul élevé et, de ce fait, réduit le nombre d'agents pouvant être utilisés dans le SMA, limitant par là même les possibilités d'utilisation des techniques adaptatives d'auto-organisation et contraignant la spécification d'une structure hiérarchique relativement figée.

Ainsi, la description d'un SMA adapté au contrôle de bioprocédés se doit d'inclure cette capacité d'apprentissage en sus des caractères génériques et adaptatifs pouvant apparaître dans les systèmes présentés dans ce chapitre.

Il est également intéressant de noter qu'en parallèle avec la problématique du contrôle de bioprocédés, l'utilisation des SMA pour résoudre des divers problèmes biologiques connaît depuis ces dernières années un certain essor. Ces approches sont en effet particulièrement adaptées aux étapes de modélisation et de simulation de phénomènes biologiques. Parmi les travaux réalisés, [Webb 2006] présente un outil de modélisation de cellules, permettant de par l'approche orientée agents d'offrir une décomposition du système plus compréhensible qu'une description classique à l'aide d'un système d'équations différentielles. Les SMA permettent également la modélisation de population de cellules. Les travaux réalisés par [Gatti 2007] proposent une architecture reposant sur des agents afin de simuler un système composé de différents types de cellules, en particulier des cellules souches. L'avantage de l'approche utilisant des SMA est ici de permettre la réalisation d'expériences virtuelles, tout en offrant une compréhension facilité des évènements s'y produisant de par la décomposition du système.

Ainsi, ce type d'applications renforce l'idée que les apports pratiques fournis par les SMA sont cohérents afin de dépasser les limites des approches de contrôle classiques, en particulier dans la décomposition des problèmes auxquels ils sont appliqués et dans leurs capacités d'adaptation.

III.5 Conclusion

Cet état de l'art des techniques de contrôle souligne les nombreuses limites de celles-ci, découlant principalement de leur manque d'adaptabilité, ainsi que d'une spécification lourde et nécessaire afin de les appliquer à un bioprocédé précis. Le tableau III.6 résume ces caractéris-

tiques.

Il apparaît donc essentiel d'arriver à concilier une approche générique capable de s'adapter à tout type de bioprocédés, avec la quantité limitée de connaissances sur le bioprocédé cible que l'on possède.

A cela s'ajoute la problématique mentionnée tout au long de l'état de l'art concernant les multiples niveaux de contrôle existant au sein d'un même problème. En effet, l'objectif de ce travail est de fournir une méthode de contrôle la plus générique possible. De ce fait, quel que soit le niveau auquel le contrôleur opère, de la régulation d'une variable isolée au contrôle global d'un bioprocédé complet, le mécanisme du contrôle en lui-même doit être identique.

De plus, l'apprentissage nécessaire que doit réaliser le système de contrôle, dû à des informations limitées sur le bioprocédé lui-même, est à prendre en compte : le système développé doit être capable de s'adapter à la dynamique d'un procédé grâce à l'étude des variables observables seules, et ne pas nécessiter un travail d'instanciation important de la part de l'utilisateur.

Le dernier point réside dans la généricité attendue, qui demande à ce que la robustesse du système développé soit assurée. Cette robustesse doit, de par la nature même des procédés biologiques, tenir compte des diverses échelles temporelles intervenant dans le déroulement du bioprocédé. C'est pourquoi la robustesse souhaitée n'est pas seulement liée à la gestion d'un bruit compris dans un intervalle précis sur les variables. Elle dépend également de la capacité du système de contrôle à s'adapter aux délais de réactions et aux retards de mesures afin d'être capable de traiter un échantillon le plus large possible de bioprocédés différents.

Ces objectifs imposent l'utilisation d'une technologie suffisamment souple et adaptative afin de développer le système de contrôle. L'utilisation d'un Système Multi-Agent (SMA) semble à cet effet pertinente. Afin de pallier les limites identifiées en partie III.3.5, il est tout à fait possible de doter un tel SMA de caractéristiques d'adaptation poussées, tout en facilitant la compréhension de son fonctionnement interne par la définition précise des comportements locaux des agents. Grâce à cette technologie, il semble alors possible de réaliser un système de contrôle robuste et adaptatif en boucle fermée qui ne reposerait pas sur l'étude de modèles du bioprocédé à contrôler.

A cela plusieurs raisons. En premier lieu, le contrôle de systèmes complexes, tels que les bioprocédés, correspond aux critères de [Parunak 2000], développés dans la partie II.2.3, définissant les problèmes pour lesquels une résolution basée sur une approche multi-agent est bénéfique. De surcroît, le paradigme de programmation orientée agent permet d'offrir une alternative aux méthodes de résolution purement mathématiques, tout en réduisant la quantité

Tableau III.6 – Comparaison des différents types de contrôle

Nom	PID	Contrôle Adaptatif	Contrôle Intelligent	SMA
Adaptabilité	- (nécessite de lourdes modifications)	+	++	+++ (dépendante du SMA)
Généricité	+	- (dépendant de la détermination du modèle)	+ (sous réserve de données d'apprentissage disponibles)	++
Position	Locale (Au niveau d'une variable)	Globale : Supervision	Aussi bien au niveau local en complément d'autres approches, que global pour la supervision	Locale + globale
Remarques	Nécessite des améliorations complexes pour s'adapter à des systèmes non-linéaires	La généricité de ces approches augmente en même temps que la complexité de leur mise en œuvre	Aspect boîte noire limitant	De nombreux SMA aux caractéristiques variées existent

d'informations sur le système à contrôler nécessaire à l'élaboration des modèles utilisés.

Ce type d'approche tend à être de plus en plus employé comme l'indiquent les techniques de contrôle présentées dans ce chapitre et reposant sur des SMA. Bien que toutes les qualités des SMA n'y soient pas entièrement mises à profit, ces applications pratiques ont permis de souligner la pertinence de ces systèmes dans le cadre du contrôle de procédés.

Toutefois, le caractère hautement cognitif des agents utilisés et les limites de leurs capacités d'apprentissage limitent la portée des approches au contrôle des systèmes pour lesquels elles ont été spécifiquement conçues.

Or, dans le cadre de cette thèse, le système chargé du contrôle doit idéalement être en mesure de fonctionner sur des bioprocédés ayant des caractéristiques et des dynamiques différentes. Afin de s'abstraire de la connaissance de ces caractéristiques, une solution capable de les apprendre devient nécessaire. Si les systèmes à contrôler sont différents, la solution doit, de plus, être générique. Enfin, la dynamique des bioprocédés est élevée et peut varier d'un système à l'autre ; par conséquent, la faculté à contrôler de tels systèmes demande des capacités d'adaptation.

Il est donc nécessaire de déterminer précisément comment concevoir un SMA répondant à l'ensemble de ces exigences ainsi que les méthodes à mettre en œvre afin de vérifier son adéquation au problème du contrôle de systèmes complexes. Le chapitre suivant détaille donc un type particulier de SMA, nommé Systèmes Multi-Agents Adaptatifs, de ses fondements théoriques jusqu'à leur confrontation à la problématique traitée dans cette thèse.

Chapitre IV

L'approche par Systèmes Multi-Agents Adaptatifs (AMAS)

La définition des prérequis nécessaires à l'élaboration d'un système multi-agent (SMA) adapté au contrôle de bioprocédés nous conduit à envisager l'utilisation de SMA capables d'adapter dynamiquement leur fonctionnalité au problème à résoudre.

Les travaux de recherche de l'équipe SMAC se focalisent sur les Systèmes Multi-Agents Adaptatifs (ou AMAS, pour Adaptive Multi-Agent Systems) dont les facultés d'adaptation reposent sur leur capacité à s'auto-organiser. Cette auto-organisation est dirigée par l'attitude coopérative [1] des agents qui maintiennent constamment des interactions dites coopératives entre eux. En changeant leurs relations, les agents modifient la fonction collective qu'ils réalisent au sein du système. Au niveau global, la fonctionnalité réalisée par le système multi-agent émerge donc des interactions réalisées au micro-niveau.

Ainsi, les contraintes d'apprentissage communes aux différents SMA utilisés pour le contrôle de procédés peuvent être gérées par le biais de l'auto-organisation des agents du système.

Ce chapitre pose les bases de l'approche adoptée pour réaliser un système de contrôle de bioprocédés en présentant d'abord les concepts liés à la théorie des AMAS ainsi que les principes qui permettent de réaliser des systèmes à fonctionnalités émergentes. Dans un second temps, la problématique du travail de thèse est étudiée relativement aux caractéristiques de base des bioprocédés qui doivent être contrôlés d'une part, ainsi que des verrous informatiques auxquels l'approche par AMAS se confronte d'autre part.

1. Le terme "coopération" n'a pas, ici, le sens classique de "collaboration" et sera précisé dans les paragraphes suivants

IV.1 La théorie des AMAS

Cette section présente les concepts fondamentaux d'émergence et de coopération avant de présenter les théorèmes liés aux systèmes multi-agents adaptatifs ainsi que les différents moyens de les mettre en œuvre.

IV.1.1 L'émergence

L'émergence est le phénomène décrivant l'apparition de structures ou de comportements au niveau macro, que les comportements des éléments du niveau micro ne suffisent pas à expliquer. Les concepts d'émergence et d'auto-organisation sont étroitement liés puisque le phénomène émergent est le résultat du collectif et que l'auto-organisation est un moyen possible d'obtenir un tel phénomène.

L'importance de l'émergence dans le cadre de l'utilisation de SMA pour le contrôle de systèmes complexes est double. D'une part, cette émergence peut apparaître au sein des systèmes à contrôler, produisant ainsi des comportements complexes que les méthodes de contrôle classiques isolées au niveau local ne permettent pas de gérer. L'émergence implique, dans ce cas, d'être en mesure de fournir un outil de contrôle capable de s'adapter aux changements de dynamique du système provoqués par les phénomènes émergents. D'autre part, l'émergence peut intervenir au sein même du SMA développé, en particulier lorsque celui-ci est auto-organisé. Cette auto-organisation peut alors conduire à l'apparition d'un comportement global du SMA qui ne peut pas être directement déduit des comportements locaux des agents le composant. Ce concept est à l'origine d'une approche informatique, nommée le calcul émergent, selon laquelle le comportement global d'un système peut être développé en se focalisant sur le développement des comportements locaux des agents. Cette décomposition du problème vise à créer des systèmes dont le comportement attendu ne peut être réalisé directement ; par exemple, à cause de sa complexité. Il devient possible de réaliser un système dont la somme des comportements locaux, définis par le concepteur, aboutira au comportement global attendu.

La notion d'émergence est sujette à une part d'interprétation non négligeable et nécessite une méthode permettant de la caractériser précisément.

Selon [Georgé 2004], caractériser un phénomène émergent consiste en la vérification des critères suivants :

– Un phénomène émergent doit apparaître au niveau global (macro) et ne doit pas être déductible directement des actions du niveau local (micro). Il est naturellement difficile

de relier le phénomène émergent aux éléments qui l'entraînent [Van de Vijver 1997].
- Le phénomène observé doit posséder une identité propre dont la cohésion dépend fortement des parties qui le constituent [Goldstein 1999]. Ce point ne contredit pas le précédent et précise seulement que le phénomène émergent doit effectivement provenir de la combinaison d'un ensemble d'actions locales présentes dans le système.
- Le phénomène émergent suit une dynamique propre, dépendant des dynamiques locales, en étant à l'origine et exerçant sur celles-ci une influence en retour [Langton 1990].

Les SMA habituellement employés dans des problématiques de contrôle, tel que l'a souligné la section III.4, reposent sur des organisations relativement statiques. Or, une organisation figée, si elle n'est pas compensée par une importante quantité de données décrivant le comportement du système à contrôler, données pouvant par exemple prendre la forme de modèles de ce système à contrôler, peut être un frein à l'utilisation d'un SMA pour contrôler un système dynamique.

L'utilisation d'un contrôle émergent apparaît comme pertinente afin de pouvoir concevoir un SMA capable de s'adapter aux différentes dynamiques des bioprocédés et ainsi, conserver une part de généricité importante. La recherche d'une méthode permettant la réalisation d'un SMA répondant à ces critères nous a donc conduit à étudier l'approche par AMAS.

IV.1.2 Objectifs de l'approche par AMAS

Un Système Multi-Agent Adaptatif (ou AMAS) hérite des caractéristiques générales des SMA présentées dans la partie II.2.1. Sa spécificité réside dans l'utilisation de l'émergence afin de fournir le comportement attendu par l'utilisateur. L'objectif de l'approche par AMAS est donc de faire le lien entre les comportements locaux d'une collection d'agents et l'émergence de la fonction globale souhaitée.

En effet, le fonctionnement d'un AMAS repose sur l'auto-organisation des agents le composant. Chacun de ces agents agit selon les informations locales qu'il perçoit et ne possède pas forcément de connaissance sur la finalité du système auquel il appartient. Les critères guidant son comportement et, par extension, sa place dans l'organisation du SMA, demeurent strictement locaux. La fonction globale d'un AMAS provient donc de la composition par auto-organisation des comportements de l'ensemble des agents le composant. De ce fait, une modification dans l'organisation de ces agents entraîne une modification de la fonction globale, et c'est cette caractéristique qui procure aux AMAS une capacité d'adaptation leur permettant de résoudre des problèmes complexes et dynamiques. La figure IV.1 illustre ce phénomène.

Le moteur de cette auto-organisation est l'attitude sociale coopérative des agents. L'idée

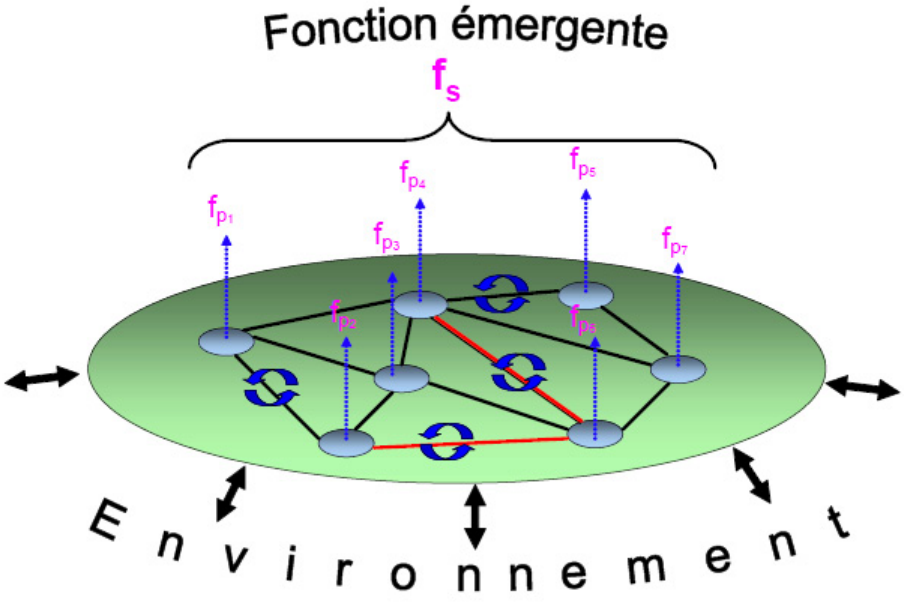

Figure IV.1 – Liens entre fonctions locales et fonction globale

sous-jacente est que le maintien de cette coopération, assuré par tous les agents du système, permet l'adaptation de sa fonction globale aux imprévus qu'il peut rencontrer. Les détails de cette coopération sont fournis dans la partie IV.1.4.

IV.1.3 Adéquation fonctionnelle

L'approche par AMAS repose sur le théorème dit de « l'adéquation fonctionnelle » [Glize 2001, Camps 1998]. L'expression adéquation fonctionnelle porte ici sur la fonction du système, c'est-à-dire, sur ce qu'un observateur extérieur au système considèrerait comme représentant le comportement du système.

L'adéquation fonctionnelle d'un système signifie donc que ledit système possède le comportement approprié afin de satisfaire l'objectif pour lequel il a été créé.

Théorème de l'Adéquation Fonctionnelle. *Pour tout système fonctionnellement adéquat, il existe au moins un système à milieu intérieur coopératif qui réalise une fonction équivalente dans le même environnement.*

La notion de milieu intérieur coopératif qualifie un SMA au sein duquel tous les agents ont des interactions coopératives c'est-à-dire qu'ils évitent des situations jugées comme étant non

coopératives au sens de la définition donnée dans la section suivante. Cette situation peut être vue comme une absence de conflit entre agents formant une organisation au sein de laquelle chacun est utile.

Axiome. *Si un système n'a aucune activité antinomique sur son environnement alors il est fonctionnellement adéquat.*

Dans cet axiome, une activité antinomique désigne une action effectuée par un agent sur son environnement dont le résultat va à l'encontre des objectifs d'autres agents du système.

Ce théorème et l'axiome associé constituent la base de l'approche par AMAS et positionnent les aspects essentiels de coopération et d'auto-organisation. Ces deux points permettent de situer l'approche par AMAS dans l'ensemble des SMA présentés dans la partie II.2.2.2 comme un système auto-organisé composé d'agents coopératifs. Ainsi, la notion de coopération est la seule contrainte à respecter pour qu'un agent puisse être intégré dans un AMAS, sans tenir compte de son niveau de cognition. La partie suivante détaille les critères définissant ce comportement coopératif.

IV.1.4 Coopération

Les agents d'un AMAS suivent un cycle de vie classique, Perception-Décision-Action, et leur comportement, comme il a été précisé précédemment, est « coopératif ». Le sens de cette coopération dépasse le cadre du simple partage de ressources ou de la collaboration. Cette coopération regroupe l'ensemble des comportements permettant à l'agent de prévenir et de résoudre les situations conflictuelles apparaissant pendant le fonctionnement du système. Ces situations sont nommées Situations Non Coopératives (SNC) et peuvent concerner chacune des étapes du cycle de vie de l'agent.

De ce fait, la coopération n'est pas limitée à la prise de décision de l'agent, mais régit également ses facultés de perception et d'action. Les différentes SNC peuvent donc être qualifiées selon la phase du cycle de vie de l'agent au cours duquel elles apparaissent. Précisons également que la notion de signal intervenant dans cette définition des SNC signifie un stimulus quelconque perçu par l'agent, que ce soit, par exemple, un message en provenance d'un autre agent ou la perception d'un élément de son environnement.

- Perception
  - Incompréhension : l'agent n'est pas capable d'extraire l'information d'un signal.
  - Ambiguïté : l'agent attribue plusieurs interprétations à un même signal.

- Décision
  - Incompétence : l'agent n'est pas capable d'exploiter pour son raisonnement l'information extraite d'un signal.
  - Improductivité : l'information extraite d'un signal ne permet de tirer aucune conclusion.
- Action
  - Concurrence : l'agent pense que son action aura les mêmes conséquences que celles résultant de l'action d'un autre agent (objectifs et compétences communs).
  - Conflit : l'agent pense que son action sera incompatible avec celle d'un de ses semblables (objectifs antagonistes).
  - Inutilité : l'agent pense que son action n'aura aucune conséquence sur le monde qui l'entoure.

La présence d'au moins une de ces SNC implique que l'agent se trouve dans une situation à laquelle il doit remédier pour diriger l'AMAS vers le fonctionnement à milieu intérieur coopératif recherché afin d'accéder à l'adéquation fonctionnelle attendue. L'auto-organisation est donc guidée par les comportements de prévention et de résolution des SNC.

Une précision demeure essentielle : le comportement coopératif des agents ne signifie en aucun cas que ceux-ci sont altruistes. Un agent ne fait pas passer les objectifs des autres agents composant le système avant les siens, mais agit pour aider le plus critique, tout en prenant garde de ne pas aggraver la situation des autres agents. De telles actions menant à la résolution des SNC tend à conduire le système vers un état d'équilibre entre les objectifs propres à chaque agent du système.

Afin de réaliser un AMAS, le concepteur doit donc spécifier de manière exhaustive l'ensemble des SNC rencontrées par les différents types d'agents du système, ainsi que les comportements appropriés pour les résoudre ou les prévenir. Les différents comportements qu'un agent peut adopter sont représentés en figure IV.2 [Bernon 2009].

Un agent ne rencontrant pas de SNC suit un comportement dit « nominal » et agit selon ses objectifs propres. Lors de la détection d'une SNC, l'agent adopte alors un comportement dit « coopératif » dont l'objectif est d'anticiper ou de résoudre cette SNC, ou d'aider l'agent le plus en difficulté. Généralement, un tel comportement coopératif peut être divisé en trois étapes distinctes :

- Le comportement d'ajustement qui consiste généralement en la modification des paramètres guidant le comportement nominal de l'agent.
- Le comportement de réorganisation, au cours duquel l'agent tente de modifier sa place

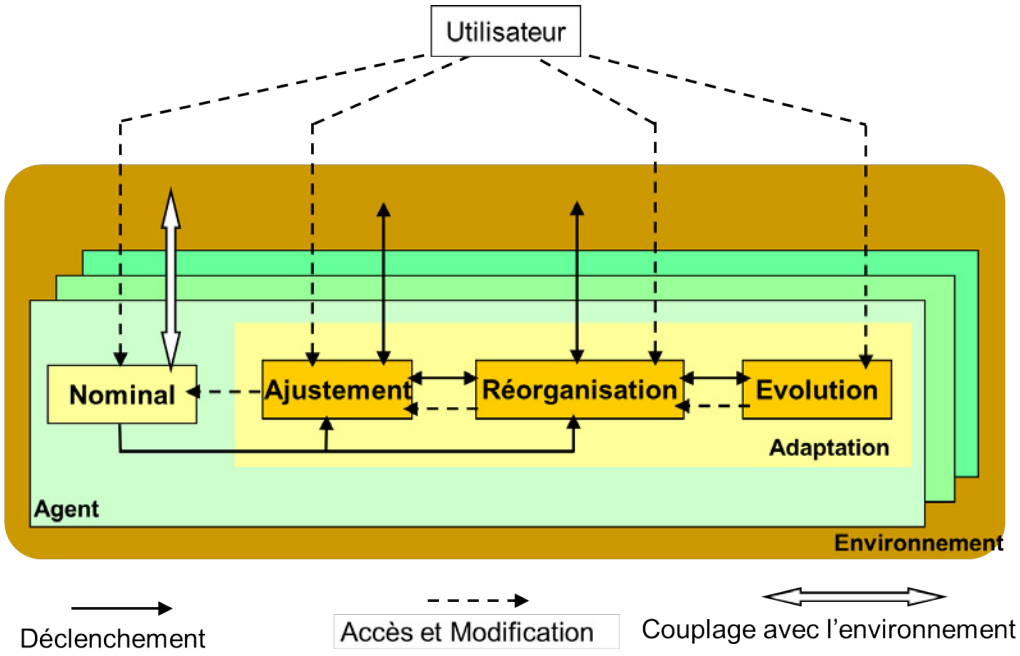

Figure IV.2 – Modèle du comportement d'un agent

dans l'organisation du système afin de rendre la fonction globale résultante adéquate. Un agent peut, par exemple, modifier les liens qu'il possède avec les agents composant son voisinage. Ces modifications affectent donc la structure de l'AMAS tout en conservant les agents existants.

– Le comportement d'évolution, qui permet à l'agent de générer de nouveaux agents ou de se détruire s'il ne se considère pas suffisamment pertinent. Cette modification entraîne une évolution de la structure de l'AMAS, par ajout ou suppression d'agents.

IV.1.5 Concevoir des AMAS

La définition des comportements conduisant à la résolution des SNC représente la principale difficulté lors de la création d'un AMAS. C'est pourquoi, des outils devaient être proposés aux développeurs d'AMAS. Dans ce sens, une méthode de conception orientée agent a été proposée par l'équipe SMAC pour guider le processus de développement d'un AMAS [Bernon 2005].

Cette méthode, nommée ADELFE [2], s'appuie sur le Rational Unified Process (RUP) pour proposer un cycle de développement logiciel en cinq phases [Picard 2004] :

2. Atelier de DEveloppement de Logiciels à Fonctionnalité Emergente

- Étude des besoins préliminaires. Cette phase permet d'établir un cahier des charges précis avec le client. Elle comprend, en particulier, la définition des limites fonctionnelles, des contraintes non fonctionnelles de l'application et la définition du vocabulaire lié au domaine d'application. Cette étape est donc indépendante de l'approche par AMAS.
- Étude des besoins finals. Cette phase s'appuie sur l'étude des besoins préliminaires et permet l'identification des entités (et non encore les agents) impliquées dans le système ainsi que la définition de l'environnement (selon la nomenclature adoptée dans [Russell 2009] : accessible, dynamique, etc.). De plus, cette phase étudie les interactions entre les différentes entités, afin de détecter celles pouvant conduire à des erreurs de coopération.
- Analyse. La phase d'analyse vérifie que les besoins exprimés sont en adéquation avec l'approche par AMAS. Les agents sont définis selon les entités détectées lors de la phase précédente et leurs interactions sont alors détaillées. Cette étape conduit à une première version de l'architecture logicielle.
- Conception. Cette phase regroupe les étapes conduisant à la conception complète des agents. Celle-ci inclut la gestion des interactions définies précédemment, ainsi que le traitement des différentes Situations Non Coopératives.
- Implantation. Cette dernière phase, issue des travaux de [Rougemaille 2008], permet de générer le code final de l'application. Un processus de transformation automatique de modèles vise à produire un code divisé en deux parties : le code comportemental des agents suivant l'approche par AMAS, et leur code opératoire respectant le modèle d'agent flexible présenté dans [Leriche 2006].

Les différentes étapes de cette méthodologie seront détaillées dans les chapitres suivants, au fur et à mesure de la description de la conception des AMAS réalisés au cours de cette thèse. Toutefois, avant de pouvoir appliquer l'approche par AMAS au contrôle de bioprocédés, il convient d'abord de définir clairement l'ensemble des problématiques qui seront abordées dans ce travail.

IV.2 Problématique de l'approche

Ces problématiques sont liées à l'existence de plusieurs facteurs affectant la qualité et la pertinence des données observées. Il convient aussi de prendre en compte les répercussions de ces facteurs sur le fonctionnement du système de contrôle.

Ces différents facteurs ont pour origine la complexité des bioprocédés telle qu'elle a été évoquée dans les chapitres précédents, à savoir, lors de la description des spécificités du contrôle

de bioprocédés au chapitre II et au cours des états de l'art du chapitre III. Cette complexité est en effet telle qu'il est impossible de décrire précisément l'intégralité des évènements se produisant à un instant donné. Ainsi, l'observateur ne possède que des fragments d'informations à partir desquels il se crée une image décrivant l'état du bioprocédé. Celle-ci, bien que pouvant se rapprocher de la réalité, en est une simplification et de ce fait, représente un modèle uniquement correct sous un ensemble d'hypothèses très précis. En conséquence, la confrontation de ce modèle avec le bioprocédé réel fait apparaître un ensemble d'imprécisions diverses.

Les paragraphes suivants détaillent les différents aspects liés à cette complexité ainsi que leurs conséquences. Les problématiques informatiques associées sont aussi présentées avant d'étudier leurs impacts sur la réalisation d'un système de contrôle basé sur l'approche par AMAS.

IV.2.1 L'influence du bruit

Le premier des facteurs étudiés est lié à la difficulté d'obtention de mesures observées de qualité. Cette difficulté a pour origine le bruit qui affecte les mesures effectuées au niveau d'un bioprocédé. Le bruit est ici considéré comme le décalage quantitatif existant entre une valeur réelle et sa valeur mesurée.

Un système de contrôle s'appuyant sur des données bruitées se trouve par conséquent confronté à un problème double. D'une part, il devient nécessaire de détecter le bruit et, d'autre part, l'impact de ce bruit sur la qualité des mesures doit être réduit autant que possible. La présence de bruit a donc pour conséquence principale d'augmenter la quantité de données nécessaire afin d'extraire un ensemble de données pertinentes sur lesquelles travailler.

Lorsque l'on observe un système aux caractéristiques connues, l'étape de détection peut, bien entendu, être triviale mais, cette tâche peut être grandement complexifiée dès lors que la connaissance de la présence de bruit devient incertaine. En particulier, lorsque l'objectif est de concevoir un système de contrôle générique, il s'avère nécessaire de réduire au maximum le besoin de données spécifiques au problème cible. Or, ces données jouent un rôle majeur dans la détection du bruit observé, que ce soit à l'aide des caractéristiques techniques des capteurs détaillant leur précision maximale ou des connaissances globales sur la forme de l'évolution attendue de la variable observée, permettant de détecter rapidement les mesures non pertinentes.

Le second aspect de la problématique liée au bruit réside dans sa gestion, c'est-à-dire dans la mise en œuvre de mécanismes aptes à diminuer au maximum son impact. Ce point se rapproche

de la détection du bruit, en cela qu'il peut être résolu, ou du moins simplifié, par la connaissance d'informations précises spécifiques au problème. Par exemple, la nature oscillatoire de l'évolution d'une variable ou encore la connaissance des valeurs attendues permettent de déterminer des mécanismes adaptés pour lisser le bruit et obtenir des mesures conformes à celles espérées. Ici encore, des méthodes génériques, conduisant à un résultat similaire, entraînent un besoin plus élevé en quantité de mesures afin de déterminer les données les plus pertinentes.

D'un point de vue informatique, le problème de la gestion du bruit consiste globalement à rechercher un équilibre entre la quantité d'informations spécifiques au problème à traiter, nécessaire pour gérer efficacement ce bruit, et la généricité attendue imposant de se reposer au minimum sur ces informations. La contrainte majeure provient donc de l'augmentation de la quantité de données nécessaire à la conception du système de contrôle pour atteindre la généricité souhaitée.

L'objectif guidant la conception d'un système de contrôle est donc de créer un système suffisamment robuste pour gérer ce bruit à la fois dans sa détection et dans son atténuation. Ainsi, il devient possible de baser le fonctionnement du système de contrôle sur les données les plus justes possibles et, de ce fait, de limiter la quantité d'informations nécessaire à la création du système de contrôle sans accroître démesurément le nombre d'informations lié au besoin de généricité.

IV.2.2 L'influence des délais

Une seconde problématique, issue conjointement de la nature dynamique du problème de contrôle et des spécificités des systèmes biologiques, réside dans l'influence du temps dans la perception des informations.

Tout comme le bruit représente un décalage quantitatif entre une valeur observée et sa valeur réelle, les délais constituent un décalage temporel entre l'état présent du système et celui représenté par les valeurs mesurées. Ces délais apparaissent principalement à trois niveaux distincts détaillés ci-après.

En premier lieu, des délais peuvent être constatés dans l'observation des mesures effectuées. La quantité alors mesurée ne représente pas la quantité exacte au moment de la mesure, mais celle présente à l'instant précédent, instant qui dépend du temps d'acquisition du capteur chargé de réaliser cette mesure.

Bien que ce délai puisse, dans un premier temps, sembler négligeable, l'hétérogénéité po-

tentielle des variables à mesurer, et donc des capteurs chargés de ces mesures, peut une fois cumulée générer un décalage suffisant pour fausser les calculs effectués à partir des valeurs mesurées. Ces décalages imposent également d'être en mesure de travailler à partir de données mises à jour de manière asynchrone.

L'influence du temps apparaît également au niveau de l'action à effectuer sur le procédé à contrôler. Entre la prise de décision de l'application d'une action et le moment où les actionneurs correspondants réalisent physiquement la modification attendue, un laps de temps existe. Généralement assez court, ce délai peut se révéler non négligeable selon le type d'action effectuée, en particulier selon la complexité de cette action et le nombre de modifications physiques qu'elle fait intervenir. De manière similaire aux délais liés à l'observation, celui sur les actions peut conduire deux actions entreprises simultanément sur deux variables différentes à être, en pratique, appliquées séquentiellement.

Enfin, le troisième cas d'apparition de délais, et également le plus complexe, réside dans l'existence d'un délai variable entre l'application d'une action et la détection de ses conséquences. Ce délai n'est pas seulement dû à la somme des délais relatifs aux actionneurs et aux capteurs, mais est directement lié à la dynamique propre du système à contrôler. En effet, dans le cas d'un système hautement dynamique, le temps de réaction du système lors de l'application d'une même modification n'est pas constant.

De plus, la complexité de la détection de ce phénomène est accentuée par l'éventuelle absence de conséquence pouvant résulter de l'application d'une action. De ce fait, la définition de critères permettant de décider si un événement n'a pas encore eu lieu ou n'aura jamais lieu est un problème dont la résolution n'est possible que par l'étude de certains phénomènes annexes fournissant le complément d'information nécessaire.

L'ensemble de ces délais est représenté sur la figure IV.3 qui met en relation l'évolution de la valeur d'une variable observable avec une action effectuée sur une variable contrôlable. Tout d'abord, une action est appliquée, ce qui entraîne un premier délai avant la modification effective du système à contrôler durant lequel ni la variable observable, ni la variable contrôlable ne laisse apparaître de changement. S'ensuit une latence variable liée à la dynamique même du système : la variable contrôlable est pour sa part effectivement modifiée, mais cette modification n'a pas encore de conséquences sur la variable observable. Enfin, la réaction a lieu, et le dernier délai apparaît au moment de la lecture des résultats à cause du temps d'acquisition du capteur chargé de mesurer la valeur attendue.

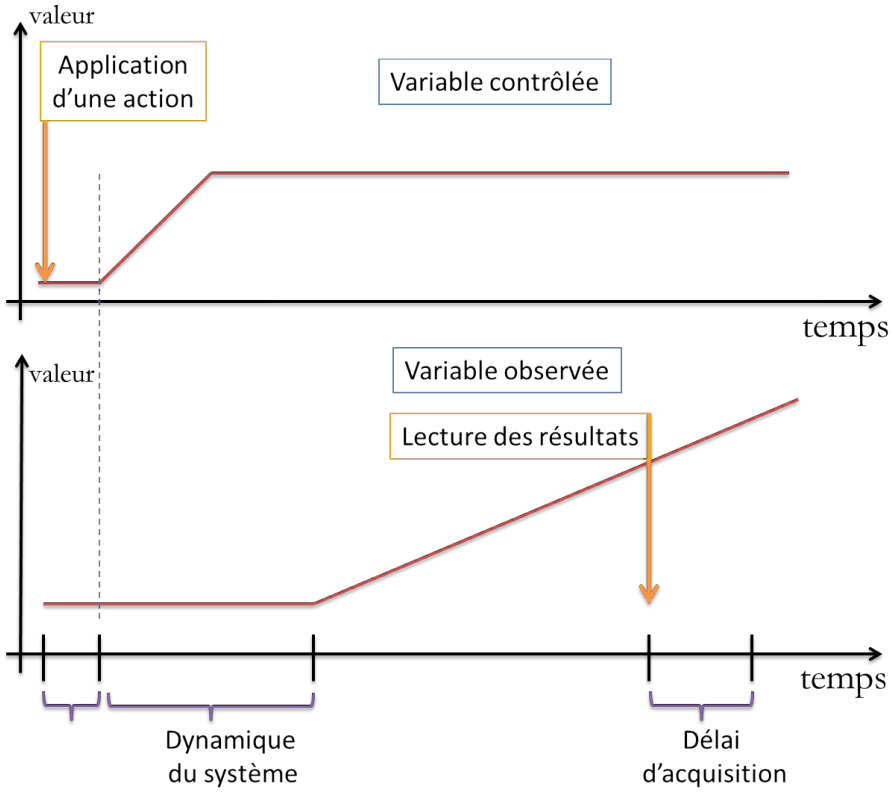

Figure IV.3 – Place des délais dans la modification et l'acquisition de données

IV.2.3 L'influence de la dynamique de l'environnement

Les paragraphes précédents ont dressé le panorama des différentes variations affectant les mesures effectuées sur le système contrôlé.

Au-delà de leur traitement au niveau de la mesure elle-même, il est nécessaire de déterminer leur impact sur le fonctionnement du système de contrôle. En effet, même sans entrer dans des considérations sur la quantité et le type de bruit présent, le système de contrôle possède déjà une information : les données dont il dispose sont potentiellement erronées. De ce fait, la conception d'un SMA fonctionnant dans un environnement bruité se doit de gérer le traitement de ce type d'informations et par extension, d'introduire dans le comportement des agents des mécanismes de révision des connaissances construites à partir de ces données bruitées.

L'impact principal de la gestion du bruit sur le comportement du système de contrôle apparaît dans ses capacités d'apprentissage. La question en résultant peut être posée ainsi : comment un système peut-il apprendre ou s'adapter à une dynamique particulière lorsque celle-ci subit un bruitage à la fois quantitatif et temporel ?

Une solution peut être apportée par l'observation des techniques habituellement employées afin de réduire ces bruits. La diversité des temps d'acquisition et de l'amplitude des bruits observés est mitigée grâce au traitement indépendant de chacune des variables d'un problème donné. De ce fait, les mécanismes de gestion du bruit à mettre en œuvre se focalisent au maximum sur l'utilisation de connaissances relatives uniquement à l'évolution souhaitée de la variable.

Nous pouvons également noter que, en raison de sa complexité, le problème de la gestion générique du bruit n'est que rarement adressé directement par les mécanismes d'apprentissage. Il est en général contourné grâce à l'utilisation de structures plus ou moins complexes représentant une connaissance spécifique du problème cible. Ces structures peuvent, par exemple, prendre la forme de courbes d'évolution attendue, de règles de fonctionnement ou d'un modèle complet du système à contrôler. Ainsi, la dynamique du système est gérée à deux niveaux distincts, avec en premier lieu une gestion des bruits au niveau des variables, puis le recours à différents modèles du système à contrôler pour déterminer les actions à effectuer.

IV.2.4 La nécessité d'un modèle du système à contrôler

Face à ces difficultés d'apprentissage, la solution la plus évidente, et également la plus appliquée, réside dans l'utilisation de modèles décrivant le comportement du système à contrôler. Cette utilisation permet de comparer les observations à une référence supposée correcte, regroupant en une seule entité la somme des connaissances sur la cible du contrôle permettant le traitement du bruit.

Cependant, ces modèles entraînent l'apparition de nouvelles contraintes. En effet, cela impose d'être en mesure de créer un tel modèle en amont du processus de contrôle, ou bien d'être capable de modéliser le système à contrôler de manière dynamique. Ces deux alternatives sont loin d'être triviales et demeurent fastidieuses à mettre en œuvre.

La création d'un modèle *a priori* impose une telle connaissance du procédé à contrôler qu'elle peut remettre en cause l'intérêt d'une approche auto-adaptative dans ce cas particulier. En effet, proposer un modèle utilisable du système à contrôler revient dans la plupart des cas à résoudre entièrement le problème du contrôle en amont : le modèle est alors un ensemble de règles dont l'application assure le maintien du procédé entre les bornes d'un comportement défini. La part d'adaptation du système de contrôle lui-même est donc réduite au minimum : celle du modèle décrivant l'ensemble des situations susceptibles d'être rencontrées.

D'un autre côté, la création dynamique d'un tel modèle regroupe les difficultés et les limites décrites au sujet du contrôle et globalement, se contente de déplacer la complexité de résolution du problème de l'étape du contrôle à celle de la modélisation. Par exemple, l'apprentissage de l'action à effectuer pour une situation donnée dans le cas du contrôle devient, au niveau de la modélisation, l'apprentissage de l'évolution de la valeur des variables pour un ensemble d'entrées données.

Une telle approche reste cependant cohérente avec la généricité attendue du contrôleur car elle repose sur l'apprentissage de la dynamique du système à contrôler. Elle ne nécessite donc pas de connaître entièrement le système cible du contrôle avant de le contrôler. Malgré cette caractéristique, une question demeure ouverte : peut-on garantir que les données obtenues au cours du processus de contrôle du procédé suffisent à établir un modèle du système cible permettant son contrôle ?

IV.2.5 Du point de vue des AMAS

Considérons maintenant les points spécifiques à la théorie des AMAS pour lesquels un développement s'avère nécessaire afin de répondre à la problématique posée.

L'idée principale guidant l'évolution d'un AMAS réside dans l'auto-organisation coopérative de ses composants permettant sa convergence vers un fonctionnement adéquat. Il est donc nécessaire, comme le décrit ADELFE, de déterminer précisément les interdépendances potentielles entre les agents composant le système.

L'aspect particulier de la problématique qui nous intéresse ici réside dans le fait que le problème du contrôle de systèmes est caractérisé par des actions à effectuer sur des éléments entre lesquels il existe une relation d'interdépendance impossible à définir *a priori*. Plus précisément, si l'on peut agir sur deux variables A et B, une action simultanée sur ces deux variables suit une relation implicite du type « dans la condition actuelle, le contrôle le plus pertinent à effectuer correspond à la composition du contrôle sur A et du contrôle sur B ». Ainsi, offrir une méthode de contrôle efficace impose d'être en mesure de synchroniser les actions sur des variables ne laissant apparaître aucun lien direct entre elles.

Or, bien que ces liens existent de fait dans l'environnement du système de contrôle, ils ne sont pas définissables en tant qu'interactions à inclure dans le SMA. La conséquence est double : d'une part, il est nécessaire de tenir compte de l'existence de ces relations et d'autre part, il est impossible de caractériser ces relations au moment de la création de l'AMAS. L'intérêt

de la problématique du contrôle pour les AMAS en particulier réside donc dans les moyens à mettre en œuvre afin d'intégrer ces liens externes à l'AMAS et non spécifiables dans le processus d'auto-organisation de l'AMAS.

IV.2.6 Du point de vue de l'informatique en général

Observons à présent du point de vue informatique l'ensemble des différents problèmes décrits dans les sections précédentes de la gestion des bruits à la modélisation.

Globalement, les notions d'hétérogénéité, de distribution et de mise à jour asynchrone liées à la gestion des décalages temporels et quantitatifs sont autant de points plaidant en faveur d'une approche orientée agent pour faciliter leur traitement. En effet, l'utilisation d'un système distribué au sein duquel chaque entité peut s'adapter selon ses objectifs propres semble cohérente avec les contraintes définies dans les paragraphes précédents : il est concevable que chaque variable apprenne indépendamment à gérer le bruit la concernant, et que les communications entre les différentes variables offrent un apport d'informations réduisant la quantité de données nécessaires à la réalisation d'une gestion du bruit efficace.

La problématique plus générale de l'apprentissage dans un environnement bruité peut conduire à une question posée en ces termes : doit-on nécessairement posséder un modèle du fonctionnement du système que l'on souhaite contrôler afin de réaliser ce contrôle ?

D'un point de vue informatique, cette question peut être schématisée par la Figure IV.4, décrivant deux types de fonctionnement potentiels :

- La méthode de contrôle usuelle (partie haute de la figure) considère un ensemble de mesures bruitées et un modèle du fonctionnement du système à contrôler, transformés par un processus T1 en un ensemble de mesures pertinentes et conduisant à une éventuelle mise à jour du modèle. Ces éléments sont alors fournis en entrée d'un processus T2 permettant d'en extraire les contrôles à réaliser. Deux étapes distinctes sont donc nécessaires afin d'obtenir le résultat attendu, et celles-ci utilisent un modèle du procédé à contrôler afin de décider du contrôle à appliquer.
- Une autre méthode de contrôle (partie basse de la figure) pourrait consister à considérer l'existence d'un processus T3, prenant en entrée uniquement des mesures bruitées, et fournissant les contrôles à effectuer. Le contrôle est alors effectué en une unique étape qui n'a ni à utiliser un modèle du système à contrôler en entrée, ni à gérer explicitement un tel modèle.

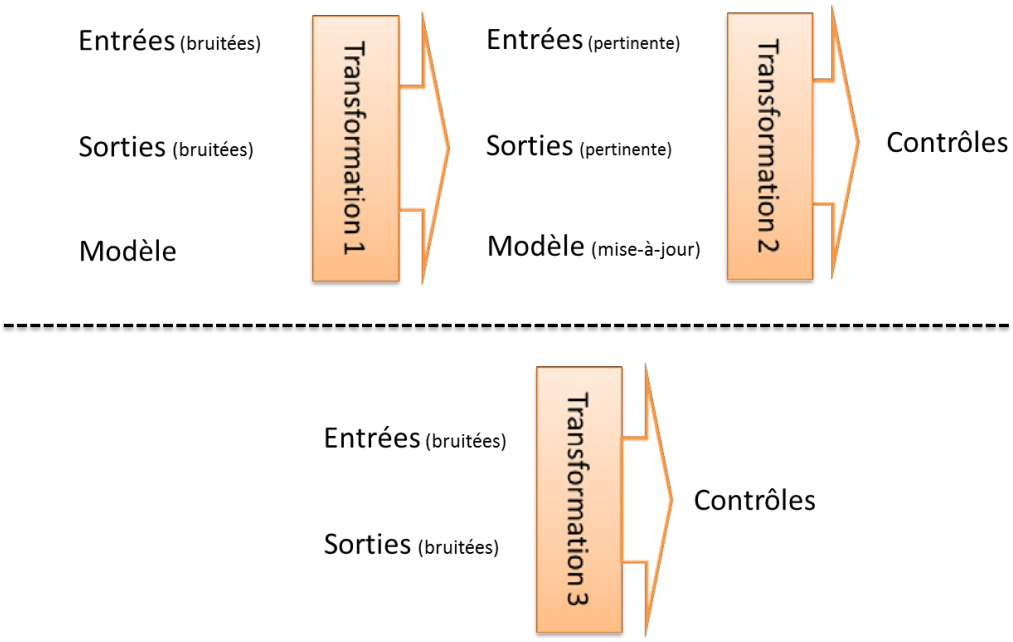

Figure IV.4 – Transformations des mesures aux contrôles

La question posée est alors de savoir si ces deux approches de contrôle sont viables dans le cas spécifique du contrôle que nous voulons traiter ici. Répondre à cette question nous a conduits à envisager les deux possibilités en concevant un système multi-agent adaptatif distinct pour chacune d'entre elles. Ces deux AMAS répondent chacun à un ensemble différent de contraintes, mais visent le même objectif : être en mesure de contrôler un système dynamique vers les objectifs définis par un utilisateur.

Ainsi, le chapitre V détaille le fonctionnement et l'application d'un AMAS chargé d'assurer les transformations T1 et T2, tandis que le chapitre VI présente un AMAS appliquant la transformation T3. Chacun de ces chapitres détaille les résultats obtenus par l'application des AMAS développés à la résolution de plusieurs problèmes de contrôle.

Chapitre V

Malachite

Ce chapitre constitue la première réponse apportée au problème du contrôle de procédés à l'aide des Systèmes Multi-Agents Adaptatifs (AMAS). Cette approche repose sur l'utilisation de modèles du système à contrôler afin de déterminer les actions à entreprendre pour satisfaire un ensemble d'objectifs définis par un utilisateur. En cela, elle suit le fonctionnement basé sur les deux transformations établies à la fin du chapitre IV.

Les hypothèses auxquelles répond le système développé sont précisées dans la première partie de ce chapitre. La structure de l'AMAS elle-même est ensuite détaillée en suivant la démarche de conception ADELFE. Enfin, l'instanciation de l'AMAS développé est réalisée sur un ensemble d'exemples dont les résultats sont commentés, avant de conclure sur le bilan des apports de l'AMAS réalisé et de discuter de sa place parmi les approches usuelles du contrôle.

V.1 Introduction

L'objectif de ce chapitre est de détailler la conception d'un AMAS s'appuyant sur le premier schéma de contrôle décrit dans le chapitre IV.2.6. Le contrôle y est effectué en s'appuyant sur deux transformations distinctes réalisées grâce à la connaissance du modèle du système à contrôler. L'AMAS réalisant le contrôle selon cette approche est nommé (arbitrairement) Malachite. Une fois sa conception décrite, les résultats obtenus pour des systèmes contrôlés particuliers sont présentés et analysés.

Le traitement du problème du contrôle est ici réalisé sous les deux hypothèses suivantes :

– On possède un ensemble de modèles décrivant le fonctionnement du système à contrôler. Ce point correspond à l'hypothèse exprimée dans le chapitre IV.2.6 concernant le type de

système de contrôle employé. Malachite suivant ce schéma, il doit donc disposer de modèles lui fournissant les informations nécessaires sur la dynamique du système à contrôler. Ces modèles peuvent être décomposés en une ou plusieurs entités nommées « pseudo-modèles », représentant un découpage plus fin des opérations de calcul réalisées par le modèle. Cette notion sera détaillée lors de la description des composants du système.

– Les entrées et sorties de ces pseudo-modèles sont observables. Il est alors possible de relier plusieurs pseudo-modèles entre eux et de déterminer les variables partagées par plusieurs pseudo-modèles.

La vérification de ces deux hypothèses implique que l'approche proposée ne tient pas compte du type de modèle utilisé et reste ainsi relativement générique. Elle peut en effet être appliquée quel que soit le type de modèles ou d'ensemble de modèles utilisés, que plusieurs pseudo-modèles puissent en être extraits ou non. De plus, la décomposition en pseudo-modèles peut être réalisée de plusieurs manières différentes pour le même ensemble de modèles d'origine.

Grâce à ces deux hypothèses, ajoutées au mécanisme général de contrôle défini dans le chapitre IV.2.6, il est possible de poser le problème du contrôle de la manière suivante. Considérons l'ensemble des modèles que l'on possède comme formant une fonction $Y = f(X)$, avec X un vecteur de valeurs d'entrées dont certaines sont modifiables par le système de contrôle et Y le vecteur des valeurs obtenues en sortie. Le calcul de Y par le biais de la fonction f et avec un vecteur X connu représente le sens dit « direct ». Contrôler un système constitué par l'ensemble de ces modèles correspond à déterminer les valeurs du vecteur X vérifiant les résultats Y définis par l'utilisateur, c'est-à-dire, à résoudre le « problème inverse ».

Les hypothèses guidant le développement de cet AMAS assurent que nous disposons des informations suffisantes pour effectuer le calcul dans le sens direct. L'objectif de Malachite est d'arriver à résoudre le problème inverse associé en se servant des informations issues de la résolution directe pour trouver les modifications à appliquer en entrée, dans la limite des actions possibles.

V.2 Application d'ADELFE

La méthode de conception ADELFE, succinctement présentée dans le chapitre IV.1.5, permet de guider le processus de développement d'un AMAS en suivant un ensemble d'étapes précis. La suite de ce chapitre détaille les activités d'ADELFE ayant soutenu la réalisation de Malachite et permettant de comprendre le fonctionnement de cet AMAS pour le contrôle.

La phase d'étude des besoins préliminaires consiste à établir et valider le cahier des charges en lien avec l'utilisateur du système à concevoir puis à définir les limites et contraintes de ce système. La définition des problématiques et des notions clés du contrôle de bioprocédés qui ont été établies dans les chapitres II et III illustrent cette phase. La description de l'application d'ADELFE pour la réalisation de Malachite va donc être détaillée à partir de la phase suivante : l'étude des besoins finals.

V.2.1 L'étude des besoins finals

Cette étape vise principalement à définir les entités présentes dans l'environnement du système de contrôle et à caractériser cet environnement.

Outre l'utilisateur du système de contrôle que l'on doit construire, selon les hypothèses de la partie V.1, deux autres entités font naturellement partie de l'environnement de Malachite : les pseudo-modèles et les variables.

La première entité présente dans l'environnement du système de contrôle est l'utilisateur de ce système. Il détermine les objectifs à atteindre et, pour cela, interagit avec le système de contrôle. Il s'agit d'une entité dite active car elle est en mesure d'effectuer elle-même des modifications sur le système. Cela se traduit dans ce cas précis par la possibilité d'ajouter, de modifier ou de supprimer des objectifs à atteindre pendant le fonctionnement du système.

L'existence des pseudo-modèles est l'hypothèse principale régissant le développement de Malachite. Ces pseudos-modèles peuvent être considérés comme des unités de calcul reliées à un ensemble de variables et permettant de calculer un ensemble de valeurs de sortie à partir d'un ensemble donné de valeurs d'entrée. Ces unités de calcul peuvent être vues comme des représentations du système à contrôler sous-jacent et donc comme des entités présentes dans l'environnement de Malachite qui vont lui permettre de matérialiser les effets du contrôle opéré.

L'étude des variables peut, quant à elle, être affinée. Il est en effet possible d'extraire deux types différents de variables, les variables contrôlables sur lesquelles une action de contrôle peut être appliquée et les variables observables pour lesquelles cette action est impossible. Ces deux types de variables partagent cependant la majorité de leurs caractéristiques. En effet, elles offrent de manière similaire la possibilité d'observer la quantité de l'élément auquel elles correspondent et peuvent également être la cible d'objectifs à atteindre de la part de l'utilisateur. Les variables forment donc le lien entre les données accessibles au système de contrôle et le

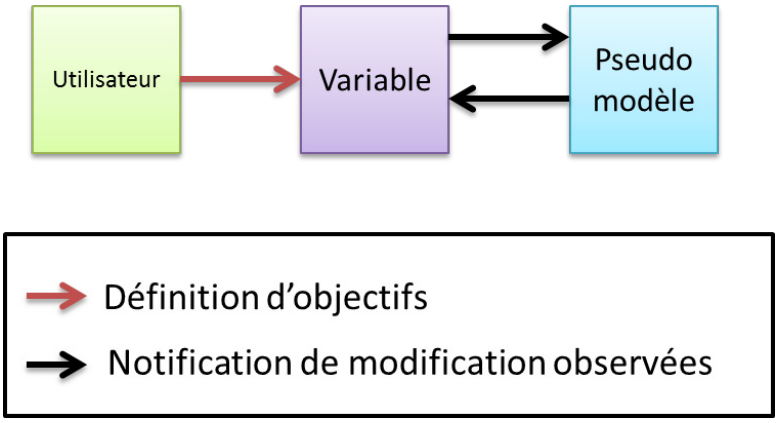

Figure V.1 – Liens entre les entités de l'environnement

système à contrôler et, à ce titre, sont des entités présentes dans l'environnement de Malachite.

Ces différentes entités appartenant au même environnement, elles maintiennent entre elles un certain nombre d'interactions. Ces dernières peuvent être détaillées en fonction de leur origine et de leur cible de la manière suivante :

- De l'utilisateur vers une variable : l'utilisateur est en mesure de fixer des objectifs sur une variable, c'est-à-dire de définir une valeur cible que la variable doit atteindre. Il impose donc des contraintes sur cette dernière. L'utilisateur peut également observer la valeur courante de la variable.
- D'une variable vers un pseudo-modèle : une variable est reliée en entrée à un ensemble de pseudo-modèles auxquels elle transmet les modifications de sa valeur qu'elle perçoit.
- D'un pseudo-modèle vers une variable : un pseudo-modèle peut transmettre le résultat du calcul qu'il effectue aux variables afin de leur notifier les modifications qu'il prévoit sur leur valeur. Le pseudo-modèle ne cherche donc pas à modifier la valeur lui-même, il transmet seulement les résultats de son calcul représentant l'éventuelle modification qui sera observée par sa sortie.

Ces trois types d'interactions permettent de définir les liens entre les différentes entités composant le système, liens illustrés par la figure V.1. Les diagrammes de séquence correspondant à ces interactions sont détaillés par les figures V.2 et V.3. Nous pouvons remarquer que l'utilisateur n'entretient aucun lien avec les pseudo-modèles et que la relation réciproque entre variable et pseudo-modèle repose sur un ensemble de notifications concernant l'évolution des valeurs mesurées

Une fois les entités de l'environnement définies, ce dernier est caractérisé à l'aide des termes issus de [Russell 2009]. L'exemple du contrôle de bioprocédés offre un environnement possédant

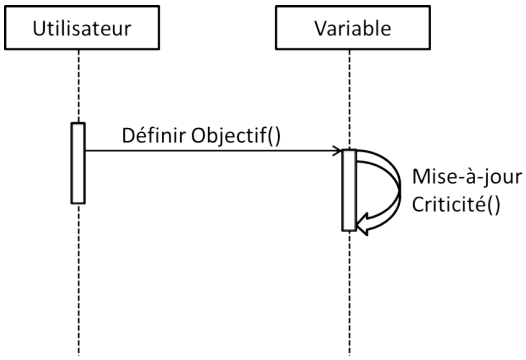

Figure V.2 – Diagramme de séquence des interactions entre utilisateur et variable

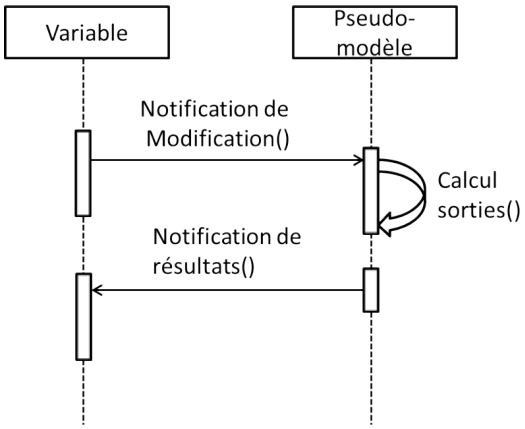

Figure V.3 – Diagramme de séquence des interactions entre pseudo-modèle et variable

les caractéristiques suivantes :

- Non accessible : ce point a été suggéré lors de la description de la problématique liée au bruit. Le contrôle de bioprocédés rend impossible l'acquisition d'information complète, exacte et à jour. De ce fait, l'environnement n'est pas considéré comme accessible.
- Non déterministe : l'application d'une action sur un bioprocédé peut entraîner des conséquences diverses, et différentes d'une simulation à l'autre. Ce phénomène permet de qualifier l'environnement de non déterministe.
- Dynamique : l'évolution de l'environnement dépend non seulement des actions du système de contrôle, mais également de la dynamique propre au procédé à contrôler. L'environnement est donc dynamique.
- Continu : le dernier point caractéristique de l'environnement est son aspect continu. Que ce soit au niveau des perceptions des mesures ou de la dynamique du système à contrôler, ces deux éléments évoluent au cours du temps et non de manière discrète.

Nous possédons à l'issu de cette phase, la liste des entités avec lesquelles le système de contrôle sera en mesure d'interagir, ainsi qu'une caractérisation de son environnement. Ces éléments constituent la base de travail qui sera enrichie et raffinée par l'application des étapes suivantes d'ADELFE.

V.2.2 L'analyse et la conception

L'étape d'analyse présente une première étape vérifiant l'adéquation du problème avec l'approche par AMAS. Celle-ci est déjà réalisée au sein du chapitre IV dans lequel les problématiques scientifiques sont développées et la pertinence de l'approche par AMAS défendue, tant au ni-

veau global que local. La phase suivante consiste donc à déterminer les agents qui composeront le système de contrôle à partir des entités définies lors de l'étape précédente. L'étude de ces entités souligne que les variables, contrôlables ou non, et les pseudo-modèles constituent les candidats à l'agentification les plus pertinents pour les raisons qui suivent.

Les variables sont des entités possédant un but local qui est la satisfaction des objectifs définis par l'utilisateur. Elles sont de plus autonomes car, dans le cas des variables contrôlables, elles peuvent entreprendre une action de modification de leur valeur propre. Les variables observables, pour leur part, déterminent si un changement perçu est pertinent ou non, ce qui souligne également leur autonomie. Enfin, les variables sont en interaction avec l'utilisateur et les différents pseudo-modèles. La satisfaction de ces trois points, objectif local, autonomie et interactions correspondent aux critères permettant d'agentifier ces entités. Avant de pouvoir être considérée comme un agent coopératif, une variable doit potentiellement pouvoir être confrontée à des échecs à la coopération (ou à des situations non coopératives). Ces situations peuvent tout d'abord se rencontrer lorsque ces entités interagissent avec d'autres, au niveau des perceptions ; par exemple, car elles ne sont pas capables de se comprendre en échangeant des messages. Toutefois, on peut envisager qu'en adoptant un protocole de communication adéquat de tels problèmes de communication ne pourront se produire et il faut alors rechercher des éventuels problèmes de coopération à un niveau différent. Si l'on considère que l'objectif d'une variable contrôlable est d'acquérir une valeur cible fixée par l'utilisateur, et que ses actions viseront à parvenir à cet objectif, ne pas pouvoir atteindre cette valeur est un échec à la coopération vis-à-vis de cet utilisateur. Bien qu'une variable seulement observable n'ait pas, sur le plan individuel, la même ambition immédiate de satisfaire l'utilisateur, elle doit tout de même concourir à un résultat collectif satisfaisant. Là encore, ne pas être capable, à un instant donné, d'atteindre une valeur acceptable pour certaines entités du système peut être vu comme un échec à la coopération envers ces entités.

En suivant un raisonnement similaire, les pseudo-modèles apparaissent également comme de potentiels agents. En effet, ces entités possèdent un but local qui est de réaliser le calcul de leurs sorties à partir de leurs entrées tout en maintenant ces valeurs à jour lorsqu'une mesure effectuée sur le système à contrôler fait apparaître une modification de la valeur d'une variable intervenant dans ce calcul. Les pseudo-modèles entretiennent également des interactions avec l'ensemble des variables composant leurs entrées et leurs sorties. La question de l'autonomie de ces entités est pour sa part plus complexe. Il est en effet envisageable de considérer les pseudo-modèles comme des ressources que les variables utilisent afin de mettre à jour leurs valeurs. Cependant, de par les hypothèses définies dans la section V.1, il est important d'insister sur le

fait que les modèles sont des entités actives jouant pour le système de contrôle le rôle d'image du procédé à contrôler. Il est alors nécessaire de positionner ces modèles, non pas comme de simples ressources, mais comme pièce centrale au fonctionnement de Malachite dont l'autonomie est héritée du système réel à contrôler. L'aspect coopératif de ces agents pseudo-modèles provient de leur nécessité d'utiliser des données en provenance d'autres agents, à savoir les agents variables, afin d'être en mesure d'effectuer leurs calculs. Ces situations peuvent donc faire intervenir des situations non coopératives si ces données ne sont pas disponibles. L'agent pseudo-modèle se retrouve donc dans l'incapacité de fournir le résultat du calcul qui lui est demandé.

Enfin, bien que pouvant répondre à la définition d'un agent, nous avons pris le parti de ne pas agentifier l'utilisateur. La raison ayant motivé ce choix réside dans le fait que l'utilisateur du système, entité unique, prend part au processus de contrôle par le seul biais de la détermination d'objectifs sur les variables. Ainsi, la totalité de ses actions peut être représentée par des modifications affectant les variables elles-mêmes. L'utilisateur en lui-même n'a pas à intervenir dans le processus de contrôle à proprement parler.

V.2.2.1 Le modèle d'agent

Suite à l'analyse des entités du système, deux types d'agents sont retenus pour être utilisés au sein de Malachite : les agents « variables » et les agents « pseudo-modèles ». Bien que présentant des caractéristiques différentes, ces agents vont s'appuyer sur une architecture interne commune représentée sur la Figure V.4 issue de [Bernon 2003]. Cette architecture est composée d'un ensemble de modules jouant les rôles suivants :

- Module de représentation : ce module représente l'image que l'agent possède de son environnement et de lui-même, c'est-à-dire l'ensemble de ses perceptions et de ses croyances.
- Module de compétences : ce module représente le comportement nominal, par opposition au comportement coopératif, de l'agent. Il s'agit de l'ensemble des compétences utilisées par la tâche effectuée par défaut.
- Module d'aptitudes : ce module représente l'ensemble des capacités permettant à l'agent de raisonner et d'agir en fonction de ses représentations et du résultat de l'application de ses compétences.
- Module de coopération : ce module représente l'ensemble des comportements permettant à l'agent d'éviter et de gérer les différentes Situations Non Coopératives (SNC, au sens AMAS du terme, tel que défini dans le chapitre IV) auxquelles il peut être confronté.

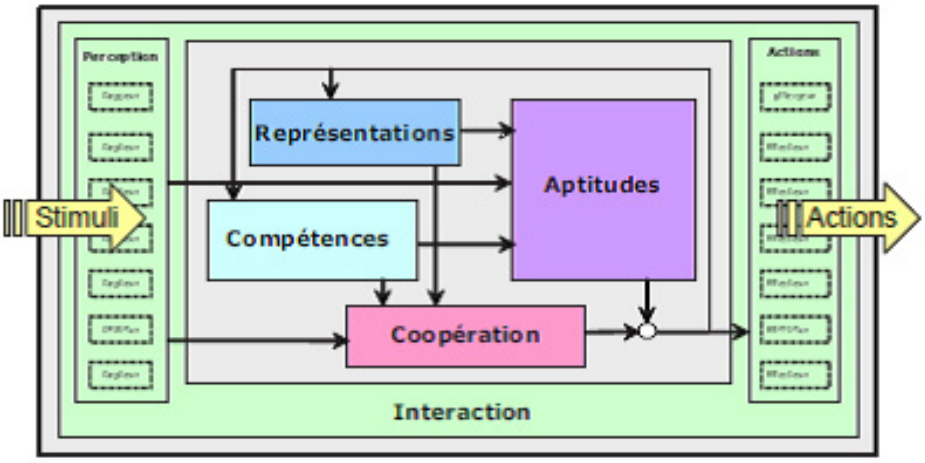

Figure V.4 – Architecture d'un agent coopératif

Bien que la majorité de ces modules soit propre à chaque agent, il est possible d'en extraire des fondements communs aux agents variables et aux agents pseudo-modèles. Ceux-ci sont, d'une part, le référentiel de représentations adoptées et, d'autre part, les aptitudes de communication mises en œuvre.

V.2.2.1.1 Le module de représentation : la criticité Afin de permettre aux agents de pouvoir échanger des informations pertinentes, il est nécessaire de leur imposer un référentiel commun. Ainsi, quel que soit leur rôle dans le SMA, les agents seront en mesure de se positionner vis-à-vis de l'état des autres agents. Cette notion commune est appelée la criticité d'un agent. Elle représente pour l'agent le degré de non-satisfaction de son objectif propre. Ainsi, être en mesure d'évaluer sa criticité propre est une condition nécessaire à la coopération entre les différents agents. Cette criticité est modélisée par une valeur réelle comprise entre 0 et 100, où 0 représente l'état de satisfaction maximal. La possibilité pour un agent d'évaluer sa criticité va permettre de déterminer son objectif propre tout en maintenant un comportement coopératif. Les méthodes d'évaluation et de calcul de cette criticité restent cependant propres à chaque type d'agent et seront, de ce fait, abordées au cours de leur description détaillée.

V.2.2.1.2 Le module d'aptitudes : la communication Les agents doivent être en mesure de communiquer pour échanger leurs représentations et ainsi obtenir une image plus juste de l'état réel du système à contrôler. En ce sens, cette communication suit les liens définis par les interactions décrites lors de l'étude des entités variables et pseudo-modèles. Cette communi-

cation est réalisée au moyen d'envois de messages et impose donc aux agents d'être en mesure de les concevoir et de les comprendre. Ces messages contiennent des informations permettant de modéliser les interactions telles que définies dans le chapitre V.2.1. Les deux catégories de messages suivantes sont utilisées par les agents :

- Requête : ce type de message contient une demande et a pour objet de déclencher une action de la part de l'agent la recevant.
- Réponse : une réponse est un message contenant la notification d'un événement considéré comme pertinent par l'agent, c'est-à-dire d'une observation que l'agent émetteur juge opportun de transmettre.

Le contenu de ces messages dépend évidemment du type d'agent le concevant, ainsi que des conditions conduisant à leur création, mais l'intégralité des échanges réalisés entre les agents ne repose que sur ces deux types de messages.

V.2.2.2 Les agents Variables

Le premier type d'agent est construit à partir des entités variables. De ce fait, les agents variables reflèteront la distinction réalisée lors de la description des entités variables en permettant la définition de deux types d'agents variables distincts : les agents variables observables et les agents variables contrôlables. Malachite comportera un agent variable pour chaque variable observable ou contrôlable accessible sur le modèle du système à contrôler.

V.2.2.2.1 Représentations L'unique représentation qu'un agent variable possède est la valeur de sa criticité. Le calcul de cette criticité peut être réalisé de deux manières distinctes. Dans la première méthode de calcul, la criticité est utilisée pour représenter un ensemble de connaissances biologiques ou physiques que l'on possède sur l'agent. Lors de sa création, une courbe représentative de sa criticité est réalisée à l'aide d'une combinaison de fonctions affines. Cette méthode est illustrée par la figure V.5 qui représente une répartition de criticité à l'aide de cinq fonctions sur un intervalle de valeurs [-2000 ; 2000]. Ce genre de définition nécessite d'être en mesure de qualifier physiquement la variable et d'associer une valeur de satisfaction à ses différentes valeurs. Par exemple, si un agent est lié à une variable désignant la température, la criticité associée peut prendre la forme d'une parabole dont les extrémités valent 100 pour les valeurs limites définies par le concepteur du bioprocédé.

La seconde méthode de calcul de criticité est appliquée lorsque l'utilisateur impose un objectif précis sur une variable. Par exemple, si l'utilisateur souhaite que la quantité de substrat

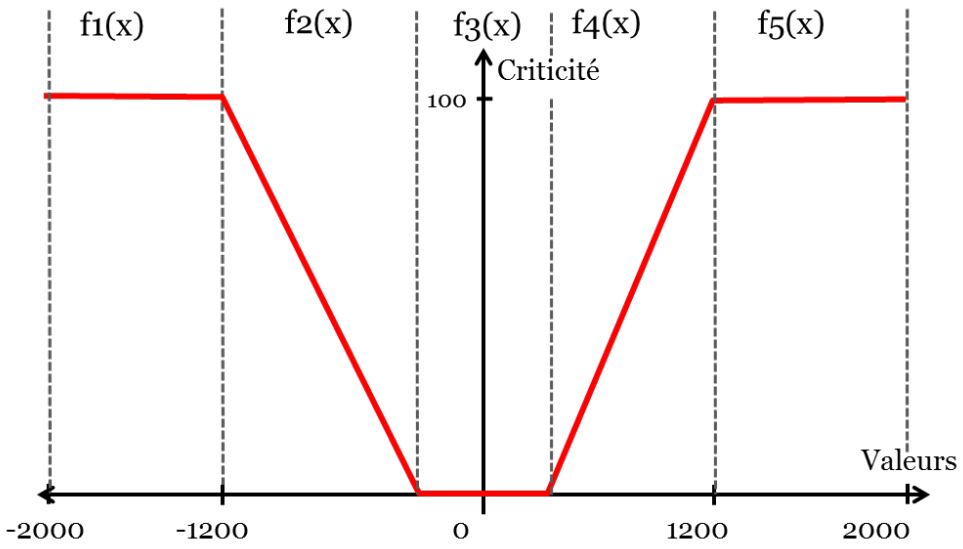

Figure V.5 – Exemple de fonction de calcul de criticité

produite par un bioprocédé soit égale à une valeur donnée, alors, la criticité de l'agent variable représentant cette quantité peut être calculée à partir de la différence existant entre sa valeur courante et sa valeur objectif. Un agent variable est donc satisfait lorsque l'objectif de l'utilisateur le concernant est atteint.

Ainsi, chaque agent variable est en mesure de calculer sa criticité propre en tenant compte des connaissances que le concepteur possède sur son évolution attendue, sur les contraintes physiques dont il est l'objet ou en s'appuyant sur les objectifs définis par l'utilisateur.

V.2.2.2.2 Compétences Les compétences d'un agent variable lui permettent d'observer l'évolution de sa valeur ; cette valeur est obtenue à l'aide de capteurs placés sur le système à contrôler. Son comportement nominal consiste à percevoir ces modifications et à notifier de ce changement les agents pseudo-modèles auxquels il est lié. Cette notification est réalisée par le biais d'un envoi de message de type réponse, tel que décrit dans la section V.2.2.1.2, contenant également la valeur de criticité courante de l'agent afin que le récepteur puisse juger des conséquences de cette modification.

V.2.2.2.3 Aptitudes La distinction entre variable observable et variable contrôlable intervient lors de la définition des aptitudes d'un agent variable. En effet, une variable observable n'impose pas d'aptitudes spécifiques alors qu'une variable contrôlable confère à l'agent la représentant la possibilité d'effectuer une action permettant de modifier sa valeur. Cette action peut

prendre des formes diverses dans l'amplitude des modifications applicables tout comme dans les délais existant avant application, mais elle est au minimum définie par une borne maximale de modification autorisée. Cette limite indique également que les actions appliquées peuvent l'être partiellement s'il est impossible à l'agent variable contrôlable de réaliser l'action totale sous ses contraintes courantes.

V.2.2.3 Les agents Pseudo-Modèles

Le second type d'agents présent dans Malachite est nommé agent pseudo-modèle. Un agent de ce type est lié à un premier ensemble d'agents variables représentant ses entrées et à un second ensemble représentant ses sorties. Il n'entretient ainsi aucun lien direct avec l'utilisateur. Malachite comporte un agent pseudo-modèle par entité pseudo-modèle extraite de l'ensemble des modèles disponibles pour représenter le comportement du système à contrôler. Le détail des modules composant ces agents est fourni dans les sections suivantes.

V.2.2.3.1 Représentation Les agents pseudo-modèles utilisent, de manière similaire aux agents variables, la notion de criticité en vue d'exprimer leur satisfaction. Cependant, une différence fondamentale est à noter : les agents pseudo-modèles ne possèdent pas de criticité propre mais héritent de la criticité maximale des agents variables auxquels ils sont liés. La raison derrière ce phénomène provient de la nature même de la criticité, représentant la satisfaction d'un agent. Les agents variables peuvent établir cette criticité car elle modélise des connaissances réelles, physiques ou mathématiques, sur le système à contrôler. Les agents pseudo-modèles représentent, quant à eux, un niveau d'abstraction supérieur, rendant la définition directe d'une criticité propre impossible. Cette criticité doit donc représenter la satisfaction de l'agent pseudo-modèle vis-à-vis de sa place actuelle dans l'organisation du SMA et ne peut donc être obtenue qu'à partir des criticités des agents voisins de l'agent pseudo-modèle, c'est-à-dire des agents variables auxquels il est lié.

V.2.2.3.2 Compétences Comme indiqué dans la description de l'entité pseudo-modèle, le rôle fondamental d'un agent pseudo-modèle est de calculer un ensemble de valeurs de sortie à partir d'un ensemble de valeurs d'entrées. La mise en œuvre de cette compétence dépend du pseudo-modèle à l'origine de l'agent, mais l'existence de cette compétence assure qu'un agent pseudo-modèle est en mesure de fournir l'ensemble des valeurs de sorties correspondant à un ensemble de valeurs d'entrées. Il est donc possible de créer un agent pseudo-modèle indiffé-

remment à partir d'un réseau de neurones, d'un système d'équations ou encore d'un ensemble d'algorithmes effectuant cette transformation des valeurs d'entrée en valeurs de sortie. L'interface permettant à un agent pseudo-modèle d'utiliser le modèle en lui-même doit donc être spécifiée selon les problèmes à traiter afin d'assurer la vérification de cette compétence.

V.2.2.3.3 Aptitudes La définition des aptitudes des agents pseudo-modèles est complexe. Ces aptitudes regroupent en effet l'ensemble des mécanismes que l'agent met en œuvre afin de pouvoir déterminer les modifications à appliquer sur ses entrées en vue d'atteindre un ensemble de valeurs de sorties donné.

Ces mécanismes se basent sur les compétences de l'agent à résoudre son propre problème direct, ainsi qu'à modifier « virtuellement », c'est-à-dire sans modification réelle sur le système à contrôler, ses entrées pour observer leur impact sur les valeurs de sorties.

L'idée guidant le processus de résolution est la suivante. Soit une équation y = f(x), avec y la valeur de sortie souhaitée et f la fonction directe appliquée par le pseudo-modèle. A l'instant t, cette équation est $y_t = f(x_t)$. Lorsque t' = t+1, on obtient alors $y_{t'} = f(x_t + \Delta x)$, où Δx représente une légère variation de x. Donc, en ajustant la valeur de Δx , nous obtenons des valeurs différentes pour $y_{t'}$. Il est alors possible pour l'agent pseudo-modèle de déterminer le Δx qui approche y de son objectif en observant le signe de $y_{t'} - y_t$. De cette manière, l'agent pseudo-modèle est en mesure de déterminer le sens des variations à appliquer sur ses entrées afin de tendre vers une valeur de sortie donnée.

Selon le problème à traiter, l'implémentation de cette aptitude peut être modifiée à l'aide, par exemple, de diverses méthodes de résolution permettant d'obtenir non pas un sens de variation mais directement une valeur solution du problème inverse local.

V.2.2.4 Les interactions entre agents et les situations non coopératives

La définition des interactions entre entités, établie dans le chapitre V.2.1, ainsi que le détail des agents permettent d'envisager un certain nombre de situations dites non coopératives (SNC). Ces SNC, définies dans la section IV.1.4, guident la création des comportements coopératifs des agents en détaillant les situations à éviter ou à résoudre pour garantir la convergence de l'AMAS vers un système à milieu intérieur coopératif. Ainsi, le comportement des agents détaillé dans la partie précédente est enrichi d'un module coopératif composé des actions décrites dans cette section.

Le fonctionnement général de Malachite peut donc conduire aux SNC suivantes, et implique l'implémentation du traitement associé :

SNC1 : La première SNC apparaît lorsque la criticité d'un agent variable devient non nulle. L'agent variable doit alors agir pour diminuer cette criticité. Cette action est réalisée à l'aide de la création d'une requête destinée aux agents pseudo-modèles pour lesquels l'agent variable en question représente une sortie. Cette requête contient la criticité de l'agent variable, ainsi qu'une valeur cible vers laquelle il souhaiterait tendre. Cette situation symbolise donc l'état d'un agent dans l'incapacité de satisfaire son objectif propre et demandant de l'aide aux agents auxquels il est lié, c'est-à-dire composant son voisinage direct.

Cette SNC est en pratique l'évènement à l'origine de la mise en œuvre des différentes actions de contrôle. Lorsque l'utilisateur impose un objectif à une variable, une modification de la fonction de criticité de l'agent variable est appliquée afin d'associer la criticité minimale pour la valeur satisfaisant l'objectif. La criticité courante de l'agent devient alors supérieure à cette criticité minimale car son état actuel est différent de l'objectif à atteindre. La hausse de la criticité entraînée par la modification de l'objectif provoque donc l'apparition de cette SNC en vue de réduire cette criticité en se rapprochant de l'objectif de l'utilisateur.

SNC2 : La seconde SNC concerne les agents pseudo-modèles. Elle apparaît lorsqu'un agent de ce type reçoit une requête (issue de la SNC précédente) à laquelle il n'est pas en mesure de répondre.

Afin de résoudre cette situation d'incompétence, l'agent pseudo-modèle va utiliser son aptitude de résolution du problème inverse afin de déterminer comment propager cette requête vers ses entrées. L'objectif est d'agir coopérativement en redirigeant cette requête vers des agents potentiellement en mesure de la satisfaire. L'étape de résolution du problème inverse effectuée par l'agent pseudo-modèle permet de transformer la requête en message compréhensible pour ses entrées, en modélisant l'impact qu'elle aurait sur leur valeur. Ce comportement conduit donc à l'émergence de la voie de communication reliant une variable demandant une modification à une action possible sur une autre variable permettant d'atteindre ce résultat.

SNC3 : La troisième SNC permet la gestion de la concurrence apparaissant lorsqu'un agent pseudo-modèle reçoit plusieurs requêtes demandant des modifications différentes et contradictoires. Dans ce cas, l'agent pseudo-modèle compare les criticités de ces requêtes, elles-mêmes représentant les criticités des agents ayant émis ces requêtes, et sélectionne la plus critique d'entre elles. Les requêtes les moins critiques sont ainsi rejetées. Ces dernières seront donc traitées ultérieurement lors de leur prochaine réception par l'agent pseudo-modèle si elles de-

viennent suffisamment critiques.

SNC4 : La dernière SNC concerne un agent variable contrôlable ne pouvant appliquer totalement une action demandée sur le système à contrôler. Cette incapacité résulte de contraintes physiques empêchant l'agent d'effectuer une modification d'amplitude suffisante, mais lui permettant cependant de l'appliquer en partie. L'agent variable va alors appliquer cette modification partielle puis notifier de ce changement les agents pseudo-modèles auxquels il est lié.

V.2.3 Bilan

Les étapes d'analyse et de conception ont permis la définition des agents composant le système de contrôle, en détaillant en particulier leur comportement ainsi que les différents niveaux de coopération qu'ils maintiennent.

Le processus général d'initialisation de Malachite est le suivant. Chaque système à contrôler entraîne tout d'abord la décomposition de ses modèles en pseudo-modèles. Les critères de décomposition sont au choix du concepteur et visent essentiellement à simplifier la mise en œuvre des mécanismes de résolution des problèmes inverses locaux aux agents pseudo-modèles. Une fois cette étape réalisée, le problème est modélisé à l'aide d'un ensemble d'agents variables (observables et contrôlable) et pseudo-modèles. L'hypothèse de la présence de modèles rend cette étape possible de manière systématique, le pire des cas étant représenté par la présence d'un seul pseudo-modèle correspondant au modèle complet du fonctionnement du système à contrôler.

Les liens entre les agents sont alors créés d'après la définition des pseudo-modèles, et l'utilisateur peut alors définir des objectifs sur une ou plusieurs variables. Ces objectifs modifient la criticité de ces dernières qui utilisent leur comportement coopératif afin de résoudre la SNC engendrée et, de ce fait, débutent la propagation d'un ensemble de messages conduisant à une série d'actions aboutissant à la satisfaction maximale des agents composant Malachite.

L'application pratique de ce fonctionnement général ainsi que les résultats obtenus sont détaillés dans les paragraphes suivants.

V.3 Application au problème de résolution d'un système d'équations

Afin de tester les contrôles réalisés par Malachite dans des conditions voisines de la problématique du contrôle d'un bioprocédé complet, plusieurs scénarios de contrôle reposant sur des systèmes composés d'équations mathématiques ont été établis. L'ensemble de ces exemples permet de valider le comportement des agents et de tester leurs capacités d'adaptations à diverses situations.

V.3.1 Description générale du problème

Le problème du contrôle d'un système d'équations consiste à agir sur un nombre limité de variables définies a priori afin de conduire un ensemble de variables vers les valeurs objectifs déterminées par l'utilisateur. Dans les scénarios suivants, le modèle du système à contrôler est représenté par le système d'équations lui-même et chacune des équations le composant jouera le rôle de pseudo-modèle, conformément à la définition de la section V.1. Les variables apparaissant dans une équation seront donc reliées au pseudo-modèle décrivant l'équation considérée. Ces équations sont exprimées sous la forme y = f(X) de manière à simplifier les liens entre agents en limitant le nombre de sorties des agents pseudo-modèles à une sortie unique. Ainsi, un même agent variable ne peut pas se trouver simultanément en entrée et en sortie d'un même agent pseudo-modèle.

La description des agents, présentée dans la section V.2.2, mentionne un certain nombre d'aptitudes dont l'instanciation dépend des éléments composant le système à contrôler. Dans le cas du contrôle d'un système d'équations, il est possible de préciser ces différents points de la manière suivante :

- Les agents variables ne subissent pas de délais lors de l'observation des valeurs des variables ou de leurs modifications. Cependant, des délais peuvent apparaître lors des temps de transmission des messages d'une extrémité de l'AMAS (variable objectif) à l'autre (variable contrôlable) selon sa topologie.
- Les capacités de résolution du problème direct des agents pseudo-modèles utilisent le calcul décrit dans l'équation du pseudo-modèle correspondant. C'est-à-dire, si un pseudo-modèle est constitué de l'équation $y = ax + b$, alors, la valeur de la sortie de l'équation, à savoir y, sera obtenue à l'aide des entrées a, x et b par application de la formule $ax + b$.

V.3.2 Contrôle d'une équation

Le premier exemple traité concerne le contrôle du système composé d'une équation unique, $y = x^2 - 2t + 1$.

Selon la composition de Malachite et les hypothèses décrites dans la description générale du problème, une telle équation conduit à la création d'un agent pseudo-modèle et de trois agents variables, t, x et y.

L'agent variable y est sélectionné pour représenter la cible des objectifs de l'utilisateur, tandis que l'agent variable x constitue l'unique agent permettant un contrôle sur le système à contrôler. L'agent y est donc un agent variable observable tandis que l'agent x est un agent variable contrôlable. Afin de complexifier le contrôle, l'agent variable t représente l'évolution d'une variable suivant une fonction linéaire inconnue des agents. La valeur de t change donc à chaque pas de simulation. De ce fait, les contrôles réalisés sur x doivent non seulement guider la valeur de y vers son objectif, mais également compenser les écarts éventuels apportés par les modifications externes induites par la présence de t. Les liens entre les agents utilisés sont représentés en Figure V.6.

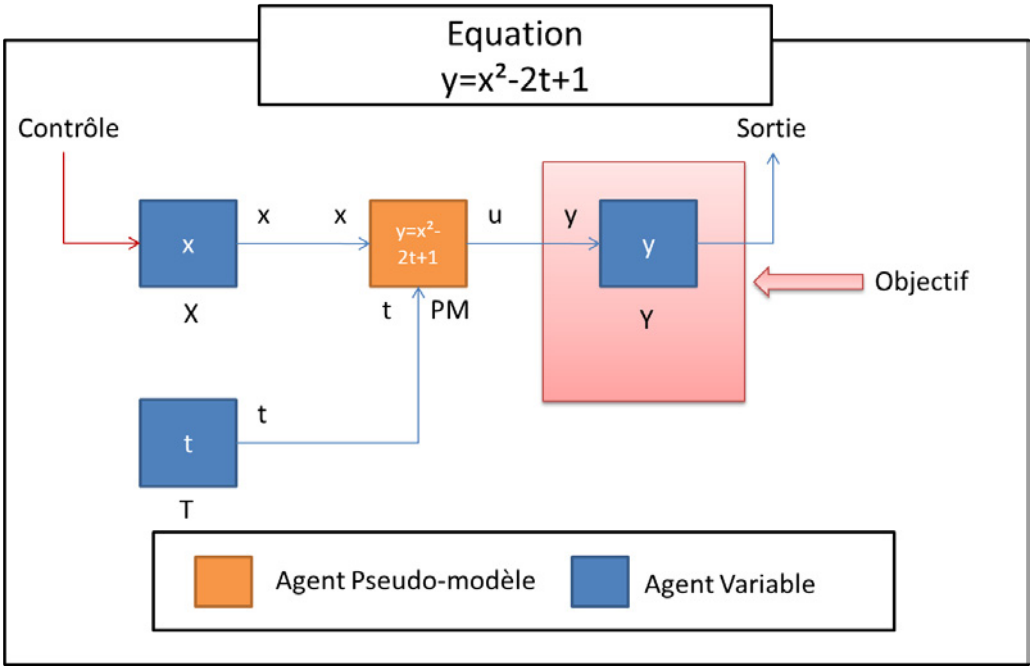

Figure V.6 – Agentification de l'équation $y = x^2 - 2t + 1$.

Le déroulement du contrôle suit le scénario suivant : les objectifs définis par l'utilisateur vont être modifiés à deux reprises au cours de la simulation afin d'observer les capacités d'adaptation

de Malachite. L'objectif initial pour l'agent variable y est 11, et cet objectif est modifié à t = 400 et t = 800 (valeurs de t indiquées en pas de simulation) pour atteindre respectivement les valeurs 1 et 6. L'ensemble de ces conditions initiales est résumé dans le Tableau V.1. La Figure V.7 présente les résultats obtenus, tout en soulignant la manière dont les actions effectuées sur la variable x permettent de compenser la dérive entraînée par la présence de la variable t.

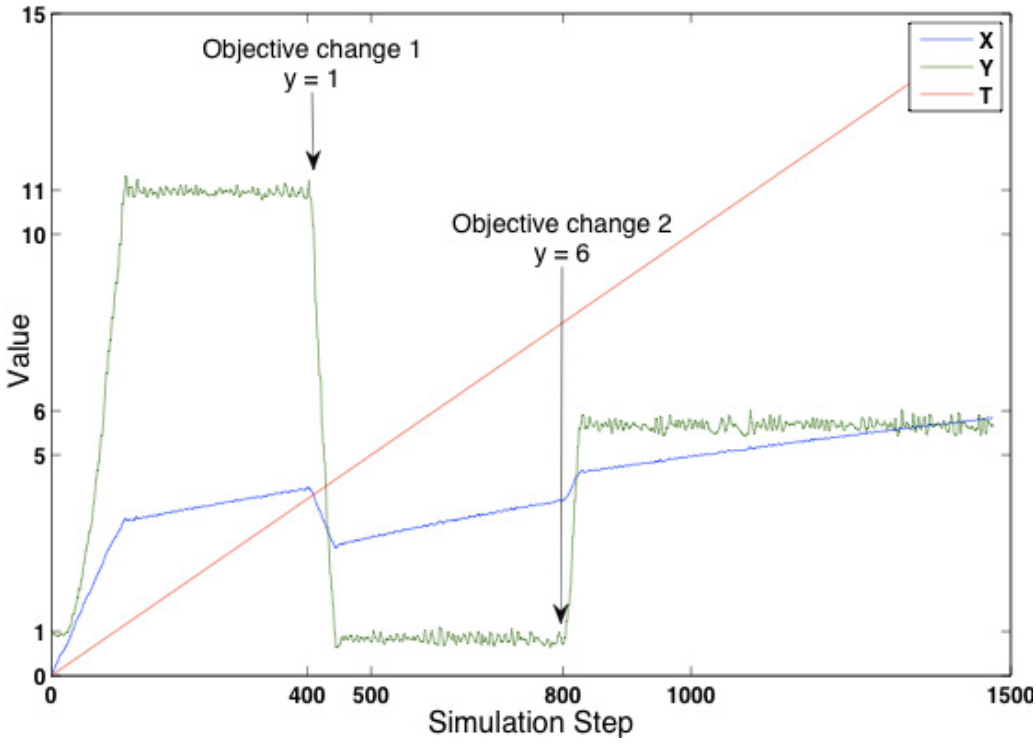

Figure V.7 – Résultats du contrôle d'une équation unique

Ces résultats mettent en évidence de nombreux points. Tout d'abord, ils permettent de souligner l'adaptation que Malachite est capable de réaliser en cours de fonctionnement afin de gérer les objectifs dynamiques fournis par l'utilisateur. Cet exemple valide également le fonctionnement des deux phases de convergence vers l'objectif et de maintien de cet objectif. La Figure V.7 illustre ce phénomène en présentant, par exemple, une première phase de convergence observable de $t = 0$ à $t = 100$. Une fois l'objectif atteint, et jusqu'à sa modification à $t = 400$, la valeur de y est maintenue grâce à une modification des actions appliquées qui sont perceptibles si l'on observe l'évolution de la forme de la courbe de x à partir de $t = 100$. Ce point souligne la pertinence de la méthode de calcul de criticité utilisée afin de modéliser les objectifs ainsi que son adéquation pour permettre au système de répercuter les changements rencontrés. Le comportement des agents, pour sa part, vérifie les critères de coopération établis durant leur

Tableau V.1 – Agents de l'exemple du contrôle d'une équation unique

Agent	Type d'agent	Valeur initiale	Objectifs (temps, valeur)	Contrôles
x	Variable	0	-	Possibles
y	Variable	1	(0,11) (400,1) (800,6)	-
t	Variable	0	-	-
$y = x^2 - 2t + 1$	Variable	0	-	-

description et tend effectivement à minimiser la criticité de ces agents. Cette criticité connaît une augmentation nette lors des changements d'objectifs opérés par l'utilisateur puis diminue progressivement au cours de la simulation.

Le second point exposé dans cet exemple est la capacité des agents à effectivement contrôler un système vers son objectif et ce, malgré la divergence apportée par la présence d'une variable non contrôlable dont la valeur évolue selon des règles inconnue et possédant un impact significatif sur le système à contrôler. L'AMAS est donc en mesure de s'adapter à la présence de variables non contrôlables, telles que peuvent l'être des mesures issues de capteurs, par exemple.

V.3.3 Contrôle d'un système d'équations

Ce second exemple vise à confronter Malachite à un problème dont la structure est bien plus complexe car faisant intervenir des boucles. Le modèle utilisé est composé d'un nombre d'équations plus important que dans l'exemple précédent ce qui a pour conséquence directe d'augmenter le nombre d'agents nécessaire pour modéliser ce problème.

Ce problème est en effet constitué d'un ensemble de trois équations faisant intervenir au total six variables distinctes dont seules deux sont contrôlables. Ces équations, décrites dans le Tableau V.2, présentent la particularité de créer un système que nous qualifierons de « bouclé » dans le sens où les variables u, x et v apparaissent à la fois en tant que solution d'une équation et en tant qu'élément de calcul de celle-ci. Ce phénomène est illustré par la structure de l'AMAS obtenue en Figure V.8, qui met en évidence la position particulière de ces trois agents variables, comparés aux agents variables y, z ou m qui n'apparaissent qu'en entrée d'une équation unique par exemple. Des contraintes ont également été rajoutées sur les actions applicables, en limitant l'amplitude de modification maximale de chaque variable par pas de simulation. Ce changement permet d'observer plus précisément les changements des modifications appliquées sur les variables contrôlables en imposant une approche plus lente des solutions.

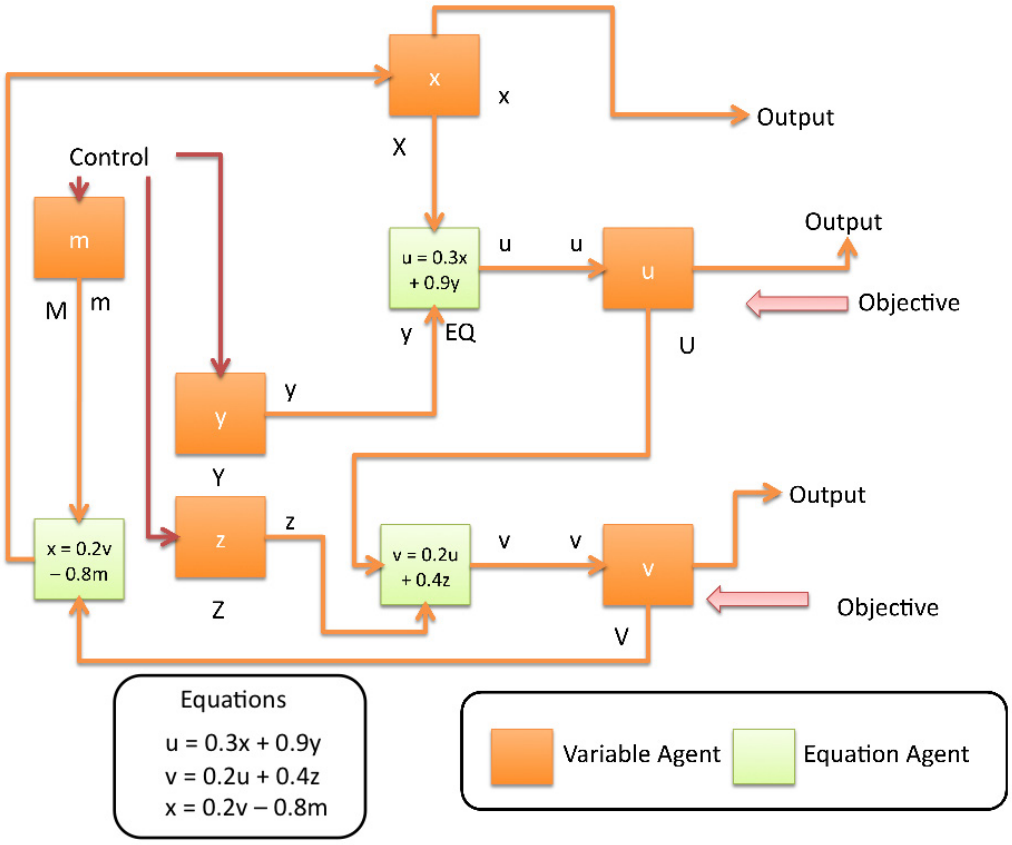

Figure V.8 – Agentification du système d'équations

Tableau V.2 – Données du système d'équations

Equation	Variables
$u = 0.3x + 0.9y$	u, x, y
$v = 0.2u + 0.4z$	v, u, x
$x = 0.2v - 0.8m$	x, v, m

Les résultats présentés en Figure V.9 permettent d'observer la gestion de la présence simultanée de plusieurs objectifs, et donc de sources de requêtes potentiellement contradictoires. Les phases de convergence et de stabilisation évoquées dans l'exemple précédent sont toujours perceptibles. Cependant, la phase de convergence présente une forme beaucoup moins linéaire, ce qui est visible en particulier sur la courbe de u. Ce phénomène provient de la recherche d'une solution à appliquer vérifiant l'ensemble des contraintes du système et donc devant vérifier les objectifs des différents agents variables. Les ajustements sont donc opérés selon les criticités des variables et conduisent à la stabilisation des objectifs.

Lorsqu'un objectif est modifié à $t = 450$, une rapide phase de convergence vers ce nouvel objectif apparaît. Le changement d'objectif ne concernant que la variable u, la variable v conserve son ancien objectif. L'agent variable V doit donc maintenir sa valeur constante tandis que l'agent u est à l'origine de requêtes visant à modifier sa valeur propre. On peut alors noter que les demandes de modifications opérées par u entraînent des modifications sur la valeur de la variable v. Celles-ci sont cependant rapidement corrigées ce qui conduit le système vers la satisfaction de l'ensemble de ces objectifs.

La phase de stabilisation finale, à partir de $t = 500$, présente une légère évolution des valeurs des différentes variables, entraînant des micro-écarts des objectifs. Ceux-ci sont dus à la gestion de la criticité par les agents variables. En effet, Malachite ne considère pas qu'une valeur soit satisfaite à moins que sa criticité soit égale à 0. Or, pour peu que le calcul de cette dernière implique des valeurs arrondies, la criticité nulle exacte ne sera jamais atteinte. De ce fait, des SNC sont générées pour demander des modifications afin d'améliorer le résultat. Ce phénomène peut être résolu à l'aide d'une prise en compte différente de la criticité en dessous d'une valeur seuil, affinant de ce fait le comportement des agents qui possèderont un comportement sensiblement différent lors des phases de convergence et de stabilisation.

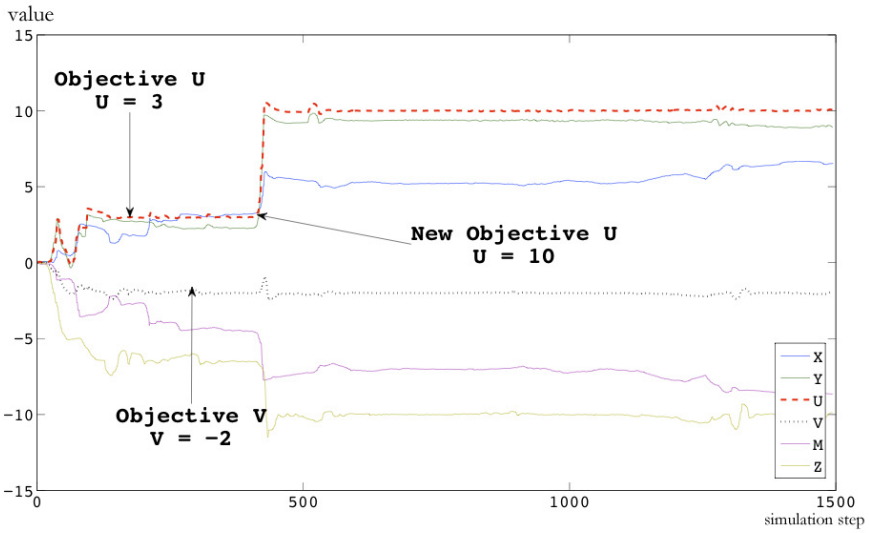

Figure V.9 – Résultats du contrôle du système d'équations décrit par le Tableau V.2

V.4 Application au problème de résolution d'un système d'équations différentielles

Malachite a été développé en visant la conception d'un système le plus générique possible. Afin de tester cette généricité, il a été employé pour contrôler un système d'équations différentielles. Ce type d'équations est communément employé afin de décrire la dynamique d'évolution des micro-organismes, et apparaît donc dans de nombreux modèles mathématiques décrivant le fonctionnement de bioprocédés réels. L'impact du changement de modèle du système à contrôler est détaillé dans la partie suivante avant de discuter les résultats obtenus.

V.4.1 Spécificités du contrôle d'équations différentielles

Bien que pouvant sembler proche des exemples précédents, l'utilisation des équations différentielles entraîne des modifications non négligeables, dans deux domaines précis. En premier lieu, des modifications sont nécessaires dans l'organisation même de l'AMAS réalisé, au sein duquel un agent variable pourra être à la fois en entrée et en sortie d'un même agent pseudo-modèle. Cependant, la principale différence liée à l'utilisation d'un système d'équations différentielles provient de la gestion des vitesses. En effet, la solution du problème direct calculée à l'aide d'une équation différentielle ne fournit pas directement la nouvelle quantité d'une variable mais une vitesse de variation de celle-ci. Les messages échangés par les agents tiennent donc compte de cette distinction et reflètent ce changement en enrichissant leur contenu d'un marqueur indiquant la nature de l'information échangée. Malgré ces différences, aucune modification n'est à apporter au comportement des agents, ni au déclenchement ou à la gestion de leurs situations non coopératives.

V.4.2 Contrôle d'un système d'équations différentielles

Cet exemple vise à démontrer deux caractéristiques particulières de Malachite. En appliquant Malachite à un modèle de nature différente du premier exemple, on cherche à mettre en évidence la généricité de cette approche, en montrant notamment que le comportement défini précédemment pour les agents peut être conservé. D'autre part, les objectifs définis par l'utilisateur dans cet exemple sont d'une complexité accrue à cause de la notion de temps qui intervient dans la manipulation des équations différentielles. Ainsi, les objectifs sont dorénavant composés d'une valeur seuil à atteindre accompagnée du temps, exprimé en pas de simulation,

Tableau V.3 – Données des équations différentielles

Equation	Variables
$u' = 0.5 * (u - 1) * v$	u, v
$v' = (1 - u) * v + 0.1a$	v, u, a

auquel l'utilisateur souhaite que cette valeur soit atteinte. La conséquence d'une telle modification réside dans la nécessité pour l'AMAS d'adapter la vitesse de modification des entrées afin d'approcher la valeur cible de la manière linéaire attendue. Ici, le choix de l'approche linéaire de l'objectif est arbitraire et ne vise pas un quelconque critère d'optimisation.

Le modèle utilisé, dans cet exemple, est composé de deux équations différentielles faisant intervenir trois variables distinctes. Ces données, détaillées dans le Tableau V.3, conduisent à la création de l'AMAS composé de deux agents pseudo-modèles représentant les deux équations et trois agents variables, AMAS qui est présenté en Figure V.10.

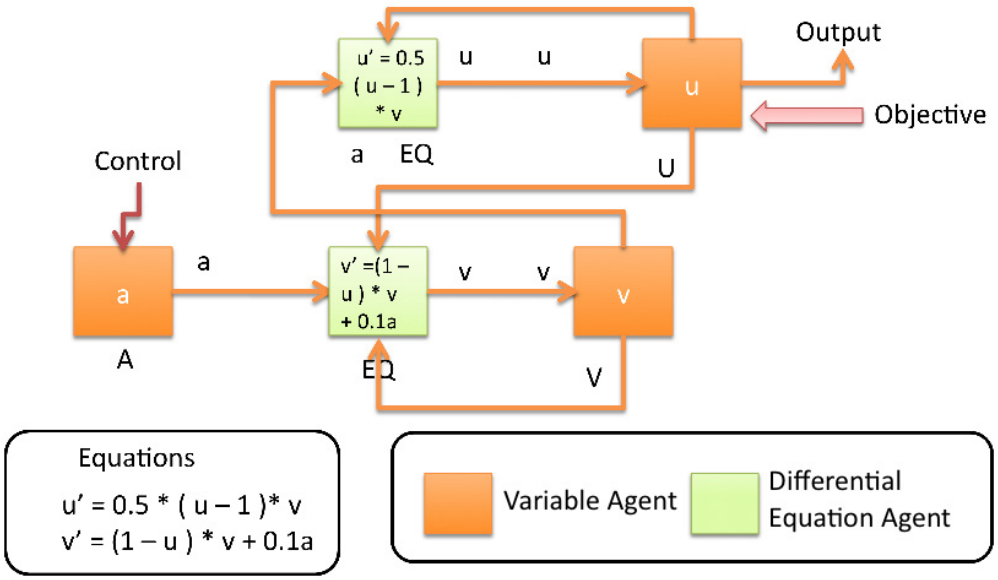

Figure V.10 – Agentification du système d'équations différentielles décrit par le Tableau V.3

Cet exemple permet d'observer la manière dont le contrôle produit s'adapte aux modifications d'objectifs définies par l'utilisateur. L'agent variable u est la cible des objectifs tandis que les seules actions possibles sont effectuées par l'agent variable a.

La Figure V.11 détaille le résultat du contrôle par rapport à la dynamique du système non contrôlé pour l'ensemble de conditions initiales décrites dans le Tableau V.4. Cette figure met en évidence les changements d'objectifs réalisés par l'utilisateur, impliquant des contraintes

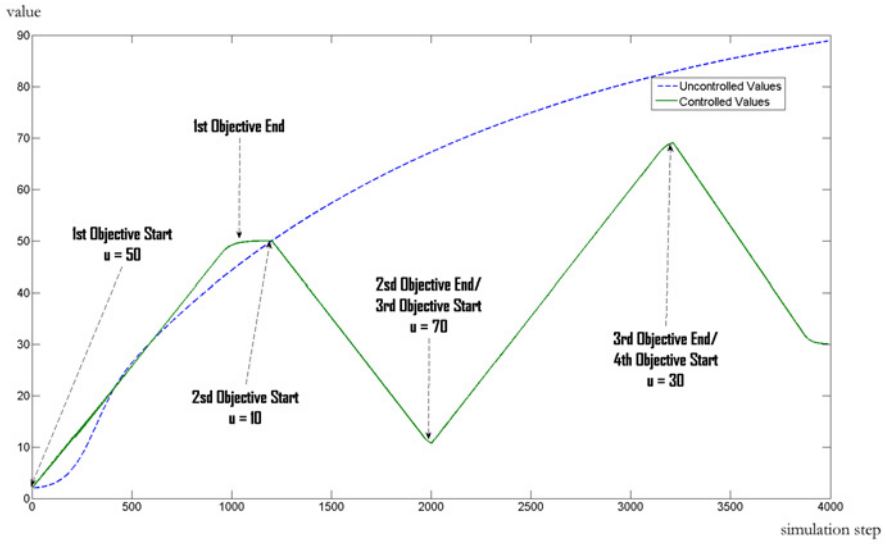

Figure V.11 – Résultat du contrôle du système d'équations différentielles décrit par le Tableau V.3

Tableau V.4 – Conditions initiales des équations différentielles

Variable	Valeur initiale	Contrôlable	Objectifs
u	3	Non	4 objectifs successifs
v	2	Non	-
a	1	Oui	-

de temps différentes pour chacun d'entre eux. En plus de la définition d'un objectif initial au démarrage de la simulation, des modifications d'objectifs (détaillées dans le Tableau V.5) sont demandées à $t = 1200$, $t = 2000$ et $t = 3200$ (en pas de simulation).

Il est immédiat de constater que l'approche linéaire imposée par la définition des objectifs est respectée comme le souligne l'aspect linéaire de la courbe résultat entre la prise en compte de l'objectif et son atteinte. La Figure V.11 laisse également remarquer une très légère diminution de la vitesse d'atteinte de l'objectif lorsque la valeur cible s'en approche afin de faciliter la stabilisation à cette valeur une fois le temps défini atteint et ainsi, minimiser l'amplitude des actions à entreprendre pour effectuer cette stabilisation.

La phase de stabilisation des exemples précédents apparaît donc également ici lorsque l'agent variable u ne possède pas d'objectif, comme le souligne le comportement observé entre $t = 1000$ et $t = 1200$.

Ces résultats soulignent la capacité de Malachite à satisfaire des objectifs prenant en compte une dimension temporelle tout en étant capable de s'appliquer sur des modèles différents. La

Tableau V.5 – Objectifs de l'agent variable u

Temps de départ	Temps limite	Objectif
0	1000	50
1200	2000	10
2000	3200	70
3200	4000	30

diversité de ces modèles peut impliquer une organisation de l'AMAS différente, comme l'a montré le passage des équations classiques aux équations différentielles, mais même celle-ci n'entraînent pas le besoin de modifier le comportement des agents pour résoudre le problème posé.

V.5 Bilan

Le travail présenté dans ce chapitre détaille donc un AMAS générique permettant de contrôler des systèmes sous l'ensemble d'hypothèses classiques décrit dans la section IV.2.6. Son application pratique a fourni des résultats pertinents sur des problèmes ayant des caractéristiques et des objectifs dynamiques de natures différentes. Ces résultats ont permis de mettre en exergue les nombreux avantages de cette approche. En premier lieu, le caractère générique de cette approche permet l'utilisation de modèles du système à contrôler exprimés sous diverses formes. Des résultats obtenus en considérant un système à contrôler décrit grâce à des équations classiques ou à l'aide d'équations différentielles ont été présentés. Ils ont permis de valider la généricité du comportement des agents au sein de deux AMAS présentant des structures différentes.

Bien entendu, cette généricité permet l'application du système de contrôle à un grand nombre de problèmes distincts. Les exemples détaillés dans ce chapitre se présentent sous la forme d'ensembles d'équations ou d'équations différentielles car ils constituent un outil fréquemment employé afin de modéliser la dynamique des bioprocédés mais ils ne sont qu'une application particulière de cette approche de contrôle. Il est en effet possible d'imaginer, en suivant la même logique de fonctionnement, un AMAS reposant sur un ensemble de réseaux de neurones. À chaque réseau de neurones correspondrait alors un agent pseudo-modèle accompagné d'agents variables tels qu'ils ont été décrits précédemment.

Outre la diversité des problèmes auxquels cette approche peut s'appliquer, il est également important de noter qu'elle demeure suffisamment ouverte pour permettre d'instancier les diffé-

rentes spécificités liées à l'application de Malachite sur des problèmes particuliers. Par exemple, l'utilisation des vitesses lors de la résolution d'équations différentielles a été possible sans modifier le comportement des agents. De même, certaines contraintes particulières telles que des valeurs limites peuvent être exprimées à l'aide des courbes de criticité des agents variables.

Enfin, les aspects de généricité et d'ouverture présentés sont renforcés par l'utilisation de cette notion de criticité comme base de la coopération entre agents. Celle-ci permet d'assurer que tous les agents présents dans l'AMAS, quelles que soient leurs instanciations, agiront de la manière coopérative souhaitée par le concepteur. En effet, la conservation du comportement coopératif des agents assure la cohérence du comportement global du système et son adéquation vis-à-vis de l'approche par AMAS.

Malachite est donc un exemple de l'utilisation des AMAS afin de modéliser un type de contrôle particulier défini dans le chapitre IV.2.6 centré sur l'utilisation de modèles.

D'un point de vue général, la principale limite de cette approche réside dans l'utilisation de modèles, et donc dans la définition même de la problématique à résoudre. Bien que celle-ci n'impose pas un type de modèle particulier, le fait même de centrer l'architecture sur un ensemble de modèles définis limite sa généricité et son fonctionnement dans des cas pour lesquels les modèles accessibles se révèlent insuffisants. De ce fait, l'objectif de s'extraire des limites présentées par les approches usuelles du contrôle n'est pas entièrement atteint.

Cette restriction bride également l'auto-organisation du système développé. En effet, bien que cette dernière apparaisse lors de la transmission des requêtes des agents vers les actions adéquates, les liens entre ceux-ci restent statiques tout au long du processus de contrôle. Cependant, l'application d'ADELFE pour la conception de Malachite n'impose pas cet aspect statique, ce dernier étant lié aux cas particuliers des modèles utilisés. Il est donc envisageable d'utiliser des modèles générés dynamiquement par un système annexe, conduisant à la création, à la modification et à la suppression dynamique de certains pseudo-modèles, tout en conservant exactement le même modèle d'agent pour déterminer les contrôles à effectuer.

La création conjointe d'un nouvel AMAS en charge de la modélisation du procédé à contrôler pourrait compléter efficacement Malachite permettant ainsi d'accroître sa généricité en limitant sa dépendance à la présence de modèles particuliers.

En ce sens, l'approche de contrôle développée dans ce chapitre peut donc être considérée comme la définition d'un ensemble de comportements complétant ceux des agents d'un SMA auto-organisé voué à la modélisation de bioprocédés. Une telle utilisation permettrait ainsi de dépasser le cadre de la problématique en réduisant grandement les informations requises sur le

bioprocédé à traiter.

Ainsi, cette première approche demeure particulièrement intéressante en tant que premier pas vers un système plus général, englobant une phase de modélisation associée, qui de ce fait n'imposera plus de prérequis sur les modèles nécessaires tout en se chargeant automatiquement de la transformation des entités du modèle généré en agents pseudo-modèles et agents variables tels que définis dans ce chapitre. Cette seconde approche est détaillée dans le chapitre suivant.

Chapitre VI

Obsidian

VI.1 Introduction

L'objectif de ce chapitre est l'étude du second schéma de contrôle proposé à la fin du chapitre IV, ainsi que la conception d'un AMAS permettant son application pratique. Le travail réalisé ici détaille donc les moyens mis en œuvre afin de se passer de modèle du système à contrôler, tout en étant capable de déterminer les actions à appliquer dans le but de conduire un système vers des objectifs définis par un utilisateur. Le système de contrôle développé dans ce chapitre est (arbitrairement) nommé Obsidian.

La problématique traitée ici partage de nombreux points communs avec celle décrite dans le chapitre V, et le processus de développement appliqué pour Malachite présente un certain nombre de caractéristiques qui seront réutilisées.

Les différences fondamentales entre ces deux systèmes résident dans les hypothèses guidant le développement de l'AMAS. Dans ce chapitre, les hypothèses suivantes forment la base du travail réalisé :

- Aucun modèle du système à contrôler n'est à disposition du système de contrôle. Ce point implique également que les connaissances issues d'un éventuel modèle ne doivent pas être intégrées de manière indirecte dans les différents agents composant Obsidian lors de sa conception.
- Le système de contrôle ne doit pas créer de modèle du système à contrôler lors de son fonctionnement. Cette contrainte a pour but d'adresser la problématique scientifique relative aux AMAS évoquée dans la section IV, en focalisant la méthode de résolution sur l'apprentissage du contrôle lui-même et non sur le fonctionnement du système à contrôler.

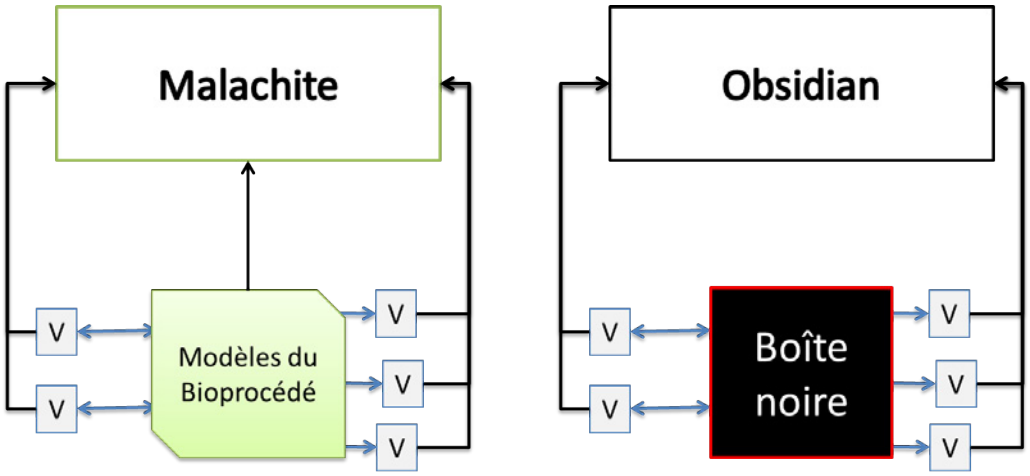

Figure VI.1 – Différences de données accessibles par Malachite et Obsidian

La présence de ces deux hypothèses impose au système de contrôle développé d'être en mesure d'apprendre par lui-même les contrôles à effectuer. Les mécanismes lui permettant de prévoir l'impact de ses modifications n'est alors plus à la charge d'un modèle extérieur au système, mais issu de son apprentissage propre.

La différence des problèmes traités par Malachite et Obsidian est résumée par la figure VI.1, soulignant cette absence de connaissance du système à contrôler.

VI.2 Application d'ADELFE

La conception du système de contrôle présenté dans ce chapitre suit la méthodologie ADELFE. La phase d'étude des besoins préliminaires étant déjà adressée dans le chapitre précédent, l'application d'ADELFE est décrite à partir de l'étude des besoins finals, première phase faisant apparaître des différences majeures entre Malachite et Obsidian.

VI.2.1 L'étude des besoins finals

L'étape d'étude des besoins finals conduisant à la conception d'Obsidian se distingue en de nombreux points vis-à-vis de celle décrite lors de la conception de Malachite. Ces spécificités dérivent des différences fondamentales entre les modèles du contrôle sur lesquels ces deux approches se basent tout comme le soulignent les hypothèses établies dans la section VI.1.

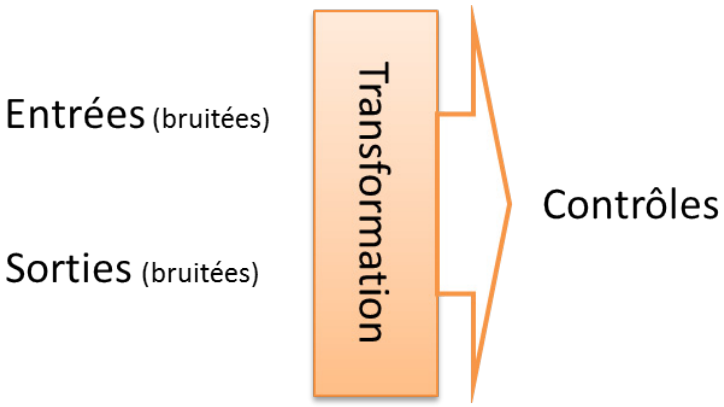

Figure VI.2 – Transformation réalisée par Obsidian

L'environnement d'Obsidian ne contient pas de modèles du système à contrôler et doit de ce fait être détaillé.

La première entité peuplant l'environnement d'Obsidian est similaire à celle présentée dans Malachite, il s'agit de l'utilisateur du système. Sa place dans l'environnement ainsi que son rôle sont similaires à ceux décrits dans le chapitre précédent ; l'utilisateur est donc une entité active déterminant les objectifs du système de contrôle, c'est-à-dire les valeurs attendues de certaines variables du système.

Les variables constituent, par hypothèse, l'ensemble des entrées du système de contrôle. La distinction physique entre variable observable et variable contrôlable est applicable, ce qui permet d'en dériver deux types d'entités. Tout comme dans le système précédent, les variables établissent le lien entre le système de contrôle et le système à contrôler.

Les entités suivantes s'écartent de celles définies dans Malachite, et participent à modéliser l'approche de contrôle spécifique employée par Obsidian. Si nous reprenons la Figure VI.2 décrivant le fonctionnement général du contrôle effectué par Obsidian, nous remarquons la présence de deux éléments dont les caractéristiques n'ont pas encore été évoquées : la transformation et le contrôle. Nous présupposons l'environnement d'Obsidian comme incluant les entités permettant de déterminer ces deux éléments.

Les contrôles retournés par la transformation sont composés d'un ensemble d'actions à appliquer sur les variables contrôlables. Il est donc possible de modéliser ce phénomène en le décomposant de la manière suivante : il existe une entité dite de contrôle pour chacune des variables contrôlables du système. Cette entité regroupe le processus décisionnel permettant de déterminer l'action à appliquer par la variable contrôlable selon l'état des variables du système. La variable contrôlable n'est donc plus en charge de cette décision, et appliquera les actions

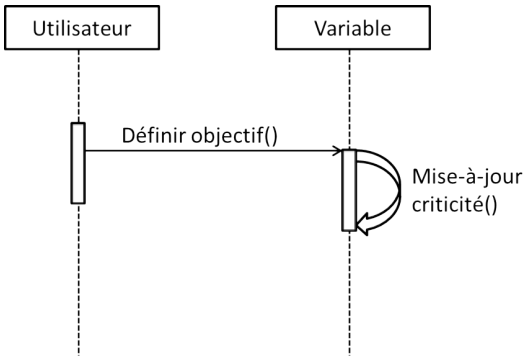

Figure VI.3 – Diagramme de séquence des interactions entre utilisateur et variable

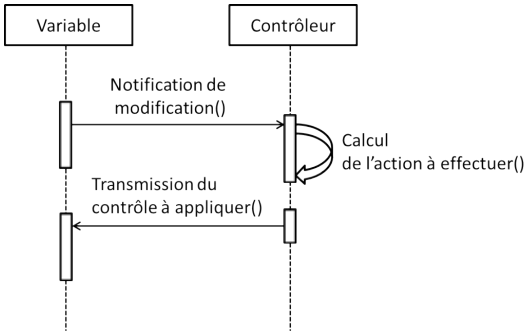

Figure VI.4 – Diagramme de séquence des interactions entre contrôleur et variable

issues des entités de contrôle.

Ainsi, les étapes de transformation et de création des actions de contrôle représentées par la figure VI.2 sont réalisées par l'ensemble des entités de contrôle, également nommées contrôleurs.

Cette nouvelle entité entretient un certain nombre de relations avec le reste de l'environnement. L'étude des interactions existant entre ces différentes entités permet d'établir les relations suivantes :

- De l'utilisateur vers une variable : l'utilisateur détermine les objectifs qui doivent être satisfaits par les différentes variables et peut accéder à leur valeur courante. De par la nature des entités variables et utilisateur, il est logique que ces interactions également présentes dans Malachite apparaissent ici.
- D'une variable vers un contrôleur : une variable est reliée à chaque contrôleur et lui transmet les notifications des évènements qu'elle considère pertinents. C'est-à-dire qu'une variable observant une modification de sa valeur va d'abord déterminer si cette mesure mérite d'être partagée avec les contrôleurs, et la leur transmettra si c'est le cas.
- D'un contrôleur vers une variable : la sortie d'un contrôleur est liée à une unique variable contrôlable à laquelle il transmet les modifications qu'elle doit appliquer sur le système à contrôler.

L'ensemble de ces interactions est résumé par les diagrammes de séquence représentés par les figures VI.3 et VI.4. Ceux-ci partagent un grand nombre de points communs avec les diagrammes établis lors de la conception de Malachite, mais il est essentiel de noter que les actions réalisées par les contrôleurs dans le diagramme VI.4 sont beaucoup plus complexes que leur équivalent à la charge des pseudo-modèles de Malachite. Cette complexité provient de l'absence de modèle pour guider le raisonnement et de la présence du processus de décision de l'action à appliquer au sein des contrôleurs.

La dernière définition à effectuer lors de l'étape d'étude des besoins finals réside dans la caractérisation de l'environnement. Celle-ci demeure similaire à celle établie pour Malachite, et définit donc un environnement non accessible, non déterministe, dynamique et continu.

VI.2.2 L'analyse et la conception

Les similitudes avec la problématique traitée par Malachite se retrouvent également au niveau de l'étape d'analyse, en particulier concernant la pertinence de l'utilisation de l'approche par AMAS. Celle-ci ayant déjà été discutée, cette section se focalise donc sur l'étape de conception.

Cette étape de conception permet de déterminer les agents qui composent Obsidian en se basant sur les entités détaillées durant l'étape d'étude des besoins finals. La conception utilise donc les définitions des entités utilisateur, variables observables et contrôlables ainsi que contrôleurs.

De par la similitude du rôle de l'utilisateur entre Malachite et Obsidian, et pour les raisons évoquées lors de la conception de Malachite, l'entité utilisateur ne conduira pas à la création d'un agent. Elle définira donc toujours ses objectifs par le biais des modifications de contraintes présentes sur les variables.

Les entités restantes susceptibles d'être agentifiées sont donc les deux types de variables et les contrôleurs.

Les variables ont, tout comme l'utilisateur, un rôle similaire dans Malachite et Obsidian, ce phénomène provient de leur proximité entre les deux modèles de contrôle soutenant ces systèmes. Ainsi, les variables utilisées ici, qu'elles soient observables ou contrôlables, partagent les mêmes caractéristiques conduisant à leur agentification. Bien qu'elles ne gèrent plus la décision de l'action à effectuer, leurs capacités de transmission d'information et de détection de sa pertinence restent à appliquer.

L'agentification des contrôleurs est, quant à lui, une étape complexe, marquant la spécificité d'Obsidian. Le rôle, défini précédemment, d'un agent fondé sur les entités contrôleurs est d'être en mesure de déterminer l'action qu'un agent variable contrôlable doit appliquer sur le système à contrôler à partir des données fournies par l'ensemble des agents variables uniquement.

Ce rôle sous-entend d'être en mesure d'associer une action avec une situation définie par les agents variables. La première étape afin de remplir ce rôle revient donc à définir la notion de situation : comment caractériser une situation à laquelle associer une action ? Une fois la

réponse à cette question apportée, il reste à définir comment définir l'action la plus pertinente à associer à la situation courante.

En résumé, un tel agent contrôle a pour objectif de répondre aux questions suivantes :

– Est-ce que l'état courant du système à contrôler correspond à une situation nouvelle ?
– Quelle action appliquer pour la situation courante ?

Il apparaît que la gestion de l'ensemble de ces comportements est une tâche trop complexe pour un seul agent, et qu'il n'existe aucune solution algorithmique simple à un tel problème. Il est donc possible de diviser ce rôle en un ensemble de deux sous-problèmes à résoudre : dans un premier temps, une étape regroupant la détermination de l'état courant du système et la gestion des situations, puis dans un second temps, le choix effectif de l'action à appliquer sur le système à contrôler.

L'idée est donc de répondre à ces questions à l'aide de la création d'un nouveau type d'agent coopératif modélisant les représentations d'un agent contrôle, et lui offrant à chaque pas de temps un ensemble de propositions d'actions applicables parmi lesquels l'agent contrôle fera son choix.

Cet agent est appelé agent contexte, en référence à son objectif qui est la représentation d'un état particulier du système à contrôler et de l'action de contrôle associée. Il est donc chargé de répondre au premier sous-problème : la définition et la gestion des différentes situations rencontrées par Obsidian.

Un agent contexte ne modélise donc pas une entité physique présente dans l'environnement d'Obsidian, mais une abstraction d'un état du système à contrôler. Il est important de noter que cette particularité ne contredit pas l'application d'ADELFE.

En effet, ce nouveau type d'agent peut être obtenu en développant le sous-problème de la gestion des situations par l'agent contrôle. Ainsi, l'agent contexte provient de la définition d'un nouvel environnement lié à la problématique d'apprentissage des situations de l'agent contrôle. Au sein de cet environnement, différent de celui dans lequel Obsidian évolue, une situation peut être représentée sous forme d'entité active cherchant à proposer une action correspondant à son état courant. L'étude de cette entité conduit à considérer son agentification.

L'aspect coopératif de l'agent contexte découle de sa position centrale entre agents variables et agents contrôles, et de son rôle d'association entre une situation et une action à appliquer. La complexité de cette tâche entraîne la nécessité du recours à un comportement coopératif, en particulier lorsque les actions proposées par un contexte conduisent à de mauvais résultats ou que plusieurs contextes différents considèrent simultanément leurs situations comme la plus

pertinente.

Une fois la problématique de la gestion des situations réalisée par l'agent contexte, l'agent contrôle à proprement parler représente alors l'aspect décisionnel des entités contrôleurs, c'est-à-dire le choix de l'action à appliquer sur le système à contrôler. L'agent contrôle est donc chargé de répondre à la deuxième sous-partie de la problématique créée par l'entité contrôleur, et utilise pour cela les résultats fournis par les agents contextes. Cette utilisation des données des agents contextes, ainsi que des éventuels conflits entre elles, imposent un comportement coopératif à l'agent contrôle.

L'étape de sélection des entités à agentifier conduit donc à la création de trois types d'agents : variable, contrôle et contexte. Bien que le premier d'entre eux soit déjà en grande partie connu de par son utilisation dans Malachite, la présence des deux autres types d'agents ainsi que leurs rôles conduisent à la formation d'un AMAS totalement différent.

Enfin, il est important de noter que les caractéristiques communes aux différents types d'agents sont similaires à celles décrites lors de la conception de Malachite, à savoir les capacités de communications par envois de messages ainsi que l'utilisation de la criticité afin de représenter la satisfaction des agents. Les parties suivantes détaillent donc les spécificités de chaque type d'agents, leur base commune étant décrite dans la partie V.2.2.1.

VI.2.2.1 Les agents Variables

Le premier type d'agent reprend un grand nombre des caractéristiques de son équivalent dans Malachite. En effet, l'entité à son origine étant similaire, il est normal que l'agentification emprunte un chemin semblable. Cependant, cette dernière présente des spécificités liées aux interactions que les agents variables entretiennent avec les nouveaux types d'agents d'Obsidian. L'ensemble des différents liens établis par les agents variables sont résumés dans la figure VI.5.

VI.2.2.1.1 Représentations Les agents variables sont très similaires à ceux utilisés par Malachite, et possèdent les mêmes mécanismes de gestion de criticité. Dans Obsidian tout comme dans Malachite, les agents variables sont l'origine des différentes criticités qui seront évaluées par l'ensemble des agents. Ces criticités sont obtenues à l'aide de la définition *a priori* de fonctions de criticité, ou par calcul par rapport à un ensemble d'objectifs fournis par l'utilisateur. L'ensemble de ces mécanismes de calcul est détaillé dans le chapitre V.2.2.2.1.

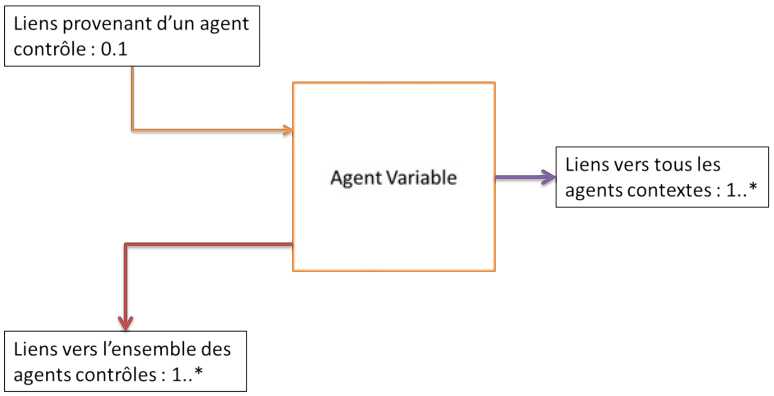

Figure VI.5 – Les liens d'un agent Variable

VI.2.2.1.2 Compétences Bien que le processus de décision permettant d'aboutir à l'action à appliquer sur le bioprocédé ne soit pas réalisé par les agents variables, ce sont quand même les agents variables contrôlables qui appliquent physiquement les modifications demandées. Les agents variables, observables comme contrôlables, ont également pour objectif de notifier les autres types d'agents du système des modifications observées concernant la valeur des variables du système à contrôler selon les liens décrits par la figure VI.5.

VI.2.2.1.3 Aptitudes La principale aptitude des agents variables réside dans les mécanismes d'atténuation du bruit qu'ils utilisent. Ceux-ci visent à transformer l'observation d'une nouvelle mesure par l'agent variable en la détection d'un phénomène nommé évènement. Un évènement est une mesure que l'agent variable considère comme étant significative, c'est-à-dire susceptible d'intéresser les agents contextes et, de ce fait, méritant d'être partagée. Par exemple, si un agent variable est chargé d'observer une variable oscillant de manière périodique, il ne transmettra pas d'évènement à chaque modification de la valeur, mais s'attardera sur l'évolution de ses tendances. Dans cet exemple précis, les variations dans l'amplitude des oscillations, ainsi que le sens de variation moyen de la valeur.

VI.2.2.2 Les agents contextes

Les agents contextes forment l'ensemble d'agents le plus complexe d'Obsidian. Ils ont pour objectif de représenter un ensemble de contrôles applicables, accompagnés de leurs répercussions prévues sur les différents agents du SMA. De ce fait, l'ensemble des agents contextes forme l'ensemble des suggestions de contrôle parmi lesquelles sera sélectionnée l'action appliquée sur le système à contrôler.

Les agents contextes voient donc leur nombre évoluer au cours du contrôle du système réel, afin d'offrir une représentation pertinente vis-à-vis de la dynamique du système à contrôler.

Il est à noter qu'un agent contexte n'est relié qu'à un seul agent contrôle, dit « associé à l'agent contexte ». Ce point s'explique par l'unicité de la cible de l'action proposée par l'agent contexte, ne concernant donc qu'une seule variable, et sera détaillé dans les paragraphes suivants. La description de ce type d'agent étant complexe, elle ne suit pas le schéma habituel en représentations, compétences et aptitudes, mais décrit progressivement les rôles de l'agent et les outils mis en œuvre pour les atteindre.

Un agent contexte dispose en effet d'une grande variété d'outils lui permettant de tenir compte de l'évolution du système réel afin de refléter au mieux une politique de contrôle pertinente pour la situation donnée. Ces différents outils permettent donc à l'agent d'évaluer sa pertinence vis-à-vis de la situation courante en répondant aux questions détaillées dans les paragraphes ci-dessous.

VI.2.2.2.1 Quand l'agent contexte représente-t-il une solution de contrôle pertinente ? La réponse à cette question réside dans l'utilisation de la notion de plages d'entrée.

L'agent contexte possède, pour définir une situation, un ensemble de données fournies par les agents variables. Ces données sont constituées des valeurs courantes des variables du système à contrôler. Or, ce système à contrôler est fortement non-linéaire, un même ensemble de valeurs de ses variables pouvant de ce fait correspondre à des états distincts nécessitant des actions de contrôle différentes. Il est donc obligatoire d'extraire des informations supplémentaires de ces données, informations apportant une meilleure précision dans la caractérisation d'une situation.

La réception des différentes valeurs des variables du système permet à l'agent contexte d'observer leur évolution, et ainsi, à l'aide de la réception de plusieurs valeurs successives, d'estimer la vitesse de variation d'une variable.

Les plages d'entrée d'un agent contexte constituent sa représentation de l'état physique de la situation qu'il représente. Elles sont composées des plages de valeurs des variables associées aux vitesses de variations de celles-ci pour lesquelles l'agent contexte pense être représentatif d'une situation précise et se considère donc pertinent.

Un agent contexte possède un nombre de plages d'entrée égal au nombre d'agents variables présents dans le système. Ainsi, l'agent contexte se positionne par rapport à la valeur de chacun des agents variables, en comparant d'une part, leur valeur réelle à l'intervalle de valeurs de sa

plage d'entrée et, d'autre part, la vitesse de variation observée de cette valeur à la vitesse correspondant à cette plage d'entrée.

L'ensemble des plages d'entrée d'un agent contexte peut être représenté de la manière détaillée par la figure VI.6, cet exemple précis faisant intervenir un agent contexte relié à trois agents variables.

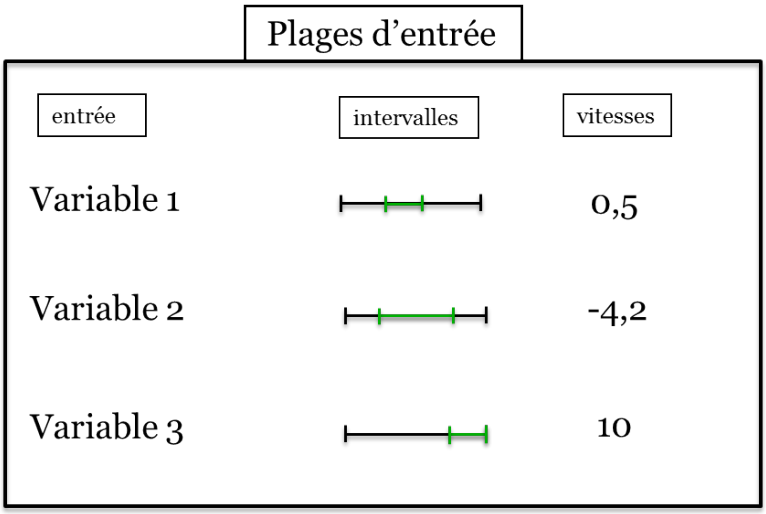

Figure VI.6 – Exemple des plages d'entrée d'un agent contexte

Les données composant les plages d'entrée peuvent être modifiées par les agents contextes eux-même. Ces modifications sont réalisées à l'aide de l'utilisation d'un outil nommé « Adaptive Value Trackers » (AVT), représentant une résolution orientée agent du problème de la recherche itérative d'une valeur susceptible d'évoluer au cours du temps à l'aide d'observations limitées du type « supérieur ou inférieur à ». Le fonctionnement des AVT est détaillé en annexe VII.3. Dans le cas des agents contextes, trois AVT sont présents pour chacune des plages d'entrée : un AVT en charge de la borne maximale de l'intervalle de valeurs, un pour la borne minimale et enfin un dernier AVT pour la vitesse de variation de la variable comprise entre les deux bornes précédentes. Ainsi, la structure de données représentée par les plages d'entrée est une structure dynamique, en mesure de s'adapter à des systèmes présentant des changements de dynamiques ou à la présence de bruit.

De nombreuses informations peuvent être dérivées de l'étude de ces plages d'entrée. Par exemple, si au cours de son cycle de vie un agent contexte voit une plage d'entrée s'étendre de la valeur minimale à la valeur maximale possible pouvant être prise par la variable correspondante, il peut être déduit que cette variable n'est pas pertinente afin de définir ce contexte : elle n'est pas suffisamment discriminante pour apporter une information caractéristique du contexte en

question.

Les plages d'entrée permettent donc aux agents contextes de déterminer quand ils doivent se manifester, c'est-à-dire lorsque l'ensemble des conditions décrivant leur pertinence sont vérifiées.

VI.2.2.2.2 Que faire lorsqu'un agent contexte se considère pertinent ? Lorsqu'un agent contexte se considère pertinent, il propose, par définition, une modification à appliquer sur le système à contrôler. Cette modification est représentée sous la forme d'une action.

Une action est une structure représentant une modification à appliquer sur une variable spécifique du système contrôlé. Cette modification peut être de complexité variable, allant de la modification instantanée d'une valeur si le système cible le permet, à un ensemble de fonctions linéaires formant une collection d'action successive à appliquer au cours du temps. La complexité de cette action dépend donc du problème auquel est instancié le système de contrôle.

En pratique, une action est représentée par un ensemble de « points d'actions » composés de deux valeurs : une vitesse et une durée. Ainsi, la forme la plus simple d'une action correspond à un point d'action unique de durée nulle, et équivaut à une modification instantanée de la valeur de la variable cible.

Cette action peut être modifiée par l'agent contexte selon les messages de rétroaction qu'il reçoit. Les modifications issues du traitement de ces messages sont réalisées à l'aide des AVT (décrits en annexe VII.3) et concernent la durée et la vitesse de chaque point d'action.

Ainsi, chaque agent contexte possède une action unique à appliquer sur une variable particulière. L'agent contexte est, de plus, capable de mettre à jour cette action afin qu'elle corresponde au mieux au contrôle permettant d'atteindre les objectifs du système.

VI.2.2.2.3 Dans quel but entreprendre cette action ? Une fois qu'un agent se considère pertinent, et propose l'application d'une action, il doit être en mesure de justifier celle-ci. Cette justification passe par l'utilisation de prévisions.

Les prévisions forment une structure de données représentant les conséquences de l'application de l'action proposée par l'agent contexte sur les criticités des agents variables. Un agent contexte possède donc autant de prévisions que d'agents variables auxquels il est lié.

Ces prévisions sont formées de manière similaire aux actions, et comportent donc deux valeurs représentant respectivement la vitesse et le temps. Ces deux informations permettent de

déterminer l'évolution de la criticité prévue sur un temps donné. Lorsque les systèmes à contrôler présentent des caractéristiques négligeant les délais d'observation, des prévisions immédiates peuvent donc être obtenues avec des valeurs de temps nulles. La figure VI.7 représente les prévisions d'un agent contexte sur l'évolution de la criticité de deux agents variables. Ainsi, cet agent contexte prévoit que son action augmentera la criticité de la l'agent variable1 tout en diminuant la criticité de l'agent variable 2.

Il est important de noter que l'impact considéré est bien celui sur les valeurs des criticités des agents variables, et non sur la valeur des variables elles-mêmes. En effet, selon la théorie des AMAS, le degré de coopération du système est ici symbolisé par la criticité. Afin d'améliorer cette coopération, et donc, de rapprocher le SMA de l'adéquation fonctionnelle, il est nécessaire d'agir en vue de réduire cette criticité.

Les fonctions de calcul des criticités peuvent être relativement complexes et dépendent des variables concernées. Par conséquent, l'observation simple de l'évolution de la valeur des agents variables ne suffit pas pour établir une tendance d'évolution de la coopération au sein du SMA. Les prévisions constituent donc la raison pour laquelle un agent contexte peut être sélectionné pour appliquer son action, ou formulé différemment, elles représentent ce en quoi s'engage un agent contexte lorsqu'il suggère une action. De ce fait, les prévisions constituent les données de base servant aux calculs des rétroactions en cas de différences entre les résultats prévus et ceux obtenus. L'adéquation de l'action de l'agent contexte sera évaluée par rapport à ses prévisions.

Tout comme les plages d'entrée et les actions, ces prévisions sont mises à jour grâce à l'utilisation des AVT.

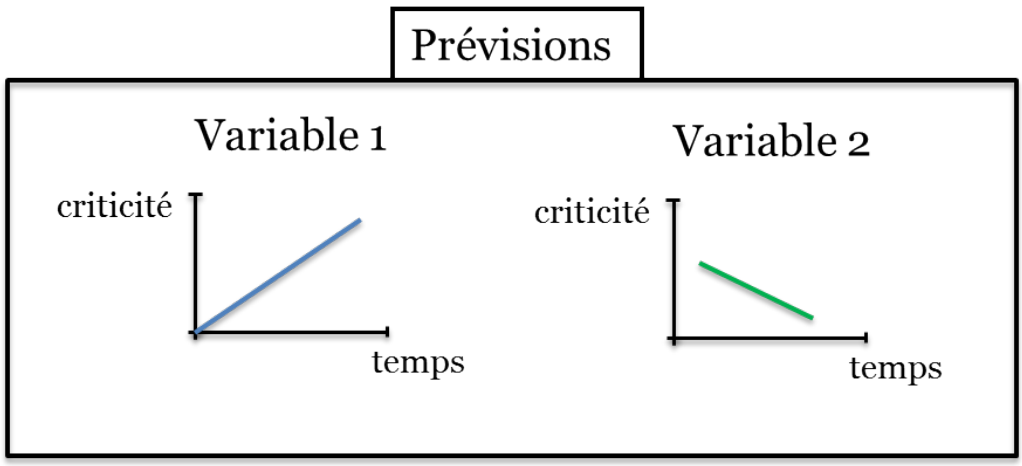

Figure VI.7 – Exemple de prévisions d'un agent contexte

VI.2.2.2.4 Dans quel état se trouve l'agent ? L'agent contexte doit posséder un mécanisme de représentation de lui-même afin d'évaluer sa pertinence. Celui-ci est réalisé à l'aide de son état interne.

L'état interne d'un agent contexte est une indication de sa situation actuelle dans l'organisation du SMA, soulignant en particulier la relation qu'il entretient avec son agent contrôle associé. Il s'agit de l'estimation que l'agent contexte réalise vis-à-vis de son utilité dans le SMA. Un agent contexte peut prendre trois états internes distincts nommés invalide, valide ou sélectionné qui définissent chacun une situation précise :

- Invalide : l'agent contexte ne se considère pas pertinent pour la situation actuelle. Ce cas apparaît lorsque les plages d'entrée d'un agent contexte ne sont pas toutes vérifiées.
- Valide : l'agent contexte se considère suffisamment pertinent pour proposer son action à son agent contrôle associé.
- Sélectionné : l'agent contexte a été reconnu comme pertinent par son agent contrôle associé, et son action est en cours d'application sur le système réel à contrôler.

La figure VI.8 illustre les liens existant entre ces différents états, en représentant en traits plein les transitions issues du comportement nominal de l'agent et en pointillés celles provenant de la réaction à certaines SNC.

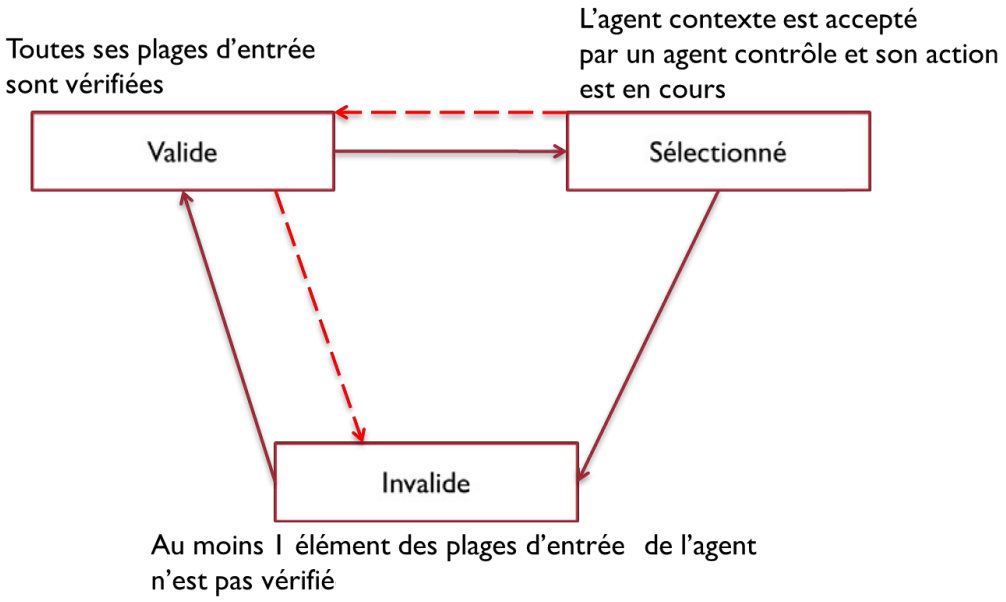

Figure VI.8 – Liens entre les différents états d'un agent Contexte

VI.2.2.2.5 Que faire si deux agents représentent la même solution ? Dans ce cas particulier, les deux agents vont tenter de fusionner en un seul et unique agent.

Ainsi, la dernière aptitude notable des agents contextes réside dans leur capacité à fusionner entre eux. L'objectif d'un tel mécanisme est de permettre à plusieurs agents se considérant proches de se regrouper pour ne former qu'un seul et unique agent. Cette proximité est évaluée à l'aide des bornes d'entrées et de l'action proposée. Ainsi, deux agents contextes dont toutes les bornes d'entrées partagent une large plage de valeurs et des vitesses similaires, tout en proposant une action du même ordre sont des candidats à la fusion. Le niveau minimal de ressemblance entre deux agents doit être défini selon le problème à traiter afin de tenir compte de la précision nécessaire pour que les caractéristique telles que les vitesses et les plages d'entrée soient suffisamment discriminantes.

L'emploi de la fusion d'agents contextes permet de généraliser des situations de contrôle ayant été construites à partir de plusieurs situations locales, représentées par des agents contextes différents.

VI.2.2.2.6 Bilan La figure VI.9 décrit l'agencement des différentes représentations et fonctionnalités présentes à l'intérieur d'un agent contexte. Ces agents sont les plus complexes d'Obsidian, mais également ceux formant le cœur de sa capacité d'adaptation. L'évolution des agents contextes représente le mécanisme mis en œuvre par Obsidian afin d'apprendre comment agir pour effectuer un contrôle efficace.

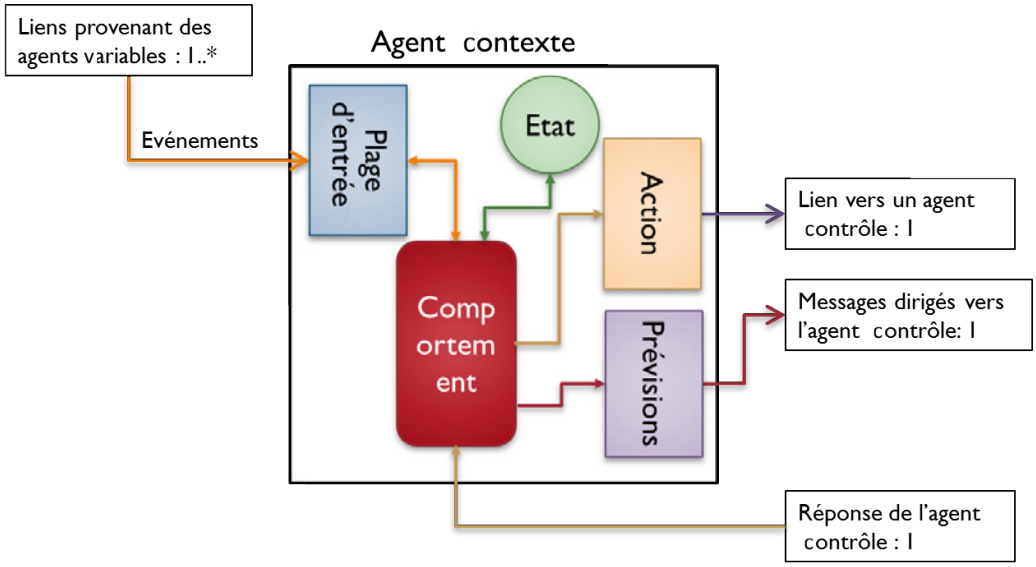

Figure VI.9 – Structure d'un agent Contexte

VI.2.2.3 Les agents contrôles

L'objectif d'un agent contrôle est de décider de l'action à appliquer sur une variable contrôlable spécifique. Cette décision est possible grâce aux propositions effectuées par les agents contextes auxquels l'agent contrôle est associé.

Le nombre d'agents contrôles présents dans le système est fixe et dépend du nombre de variables sur lesquelles une modification est possible sur le système réel à contrôler. Pour chacune de ces variables, un agent contrôle sera créé et lui sera associé. C'est-à-dire que chaque action sélectionnée par l'agent contrôle concernera uniquement cette variable associée.

VI.2.2.3.1 Représentations Les agents contrôle manipulent les criticités afin de guider leur comportements. Ces agents n'ont cependant pas de criticité propre, mais agissent en fonction des criticités que les différents agents variables leur transmettent.

Les représentations des agents contrôles comportent, au-delà des données transmises par les agents variables, les suggestions d'actions émises par les agents contextes. De ce fait, les agents contrôles possèdent une liste d'actions potentielles accompagnées des prévisions de leurs conséquences leur permettant de guider leur comportement.

VI.2.2.3.2 Compétences Le comportement d'un agent contrôle ne se résume pas à une sélection directe des agents contextes présentant les meilleures prévisions. Un agent contrôle possède également une visibilité de l'évolution des différents agents variables du système, et donc de l'évolution de la criticité de ce dernier. Son rôle est donc d'évaluer (par rapport à l'état courant du système et à la criticité des différents agents variables) quelle action proposée par un agent contexte est la plus pertinente, c'est-à-dire, celle qui répond au mieux au critère de diminution de la criticité maximale du système. A chaque pas de temps de la simulation, un agent contrôle ne sélectionne qu'un seul agent contexte et ce, même s'il a le choix entre plusieurs d'entre eux.

VI.2.2.3.3 Aptitudes Comme indiqué lors de l'agentification des entités, les agents contextes proviennent de la résolution d'un sous-problème propre aux entités contrôleurs. De par sa conception, un agent contexte peut être considéré comme appartenant à une entité contrôleur particulière et par extension, lié à un agent contrôle spécifique. De ce fait, un agent contrôle est en mesure de créer de nouveaux agents contextes lorsque certaines situations non coopératives sont rencontrées tel que le décrira le paragraphe suivant.

L'idée derrière cette aptitude est de permettre aux agents contrôles de réagir à des situations pour lesquelles aucun contexte n'est pertinent. Ce cas se présente, par exemple, lorsqu'Obsidian doit contrôler un système sur lequel il ne possède aucune information, c'est-à-dire qu'il n'a créé aucun agent contexte associé aux situations auxquelles il est susceptible d'être confronté.

VI.2.2.4 Les interactions entre agents et les situations non coopératives

La structure d'Obsidian présente plusieurs couches d'agents. Un exemple d'un tel AMAS se trouve en figure VI.10. Cet AMAS à vocation purement illustrative fait intervenir trois agents variables, parmi lesquels deux agents variables contrôlables, et deux agents contrôles liés respectivement à trois et deux agents contextes.

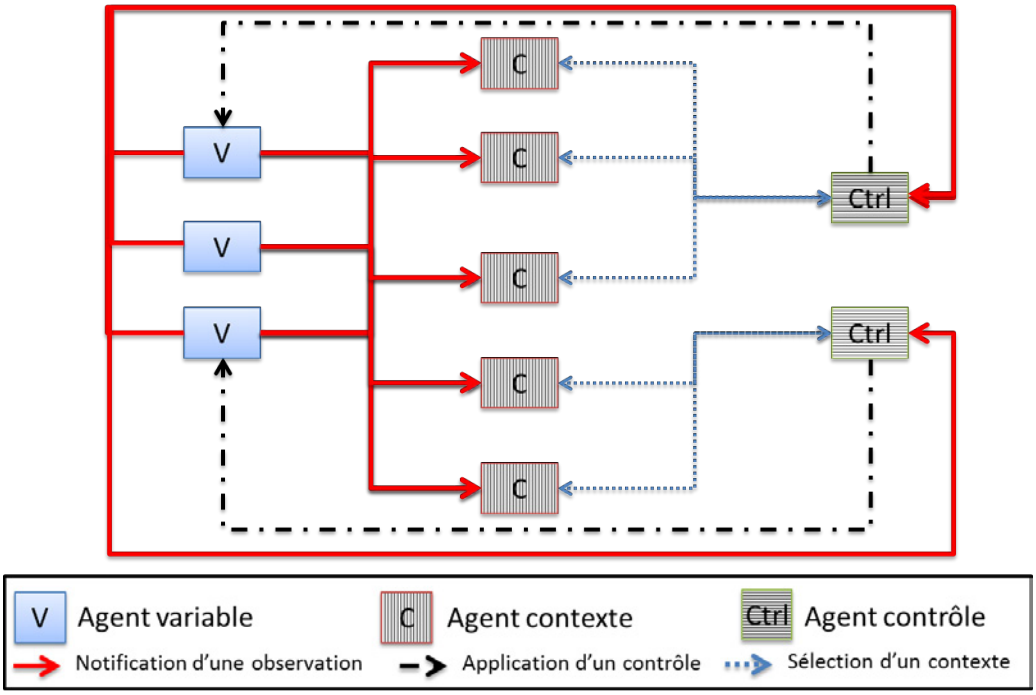

Figure VI.10 – Exemple de la structure d'Obsidian

La complexité de la structure d'Obsidian, composée de plusieurs couches d'agents, implique un nombre élevé d'interactions entre les agents parmi lesquelles il est possible d'extraire les SNC suivantes, présentées accompagnées du comportement coopératif qui leur est associé dans les paragraphes suivants.

SNC1 : Un agent contexte devient valide mais s'aperçoit à l'aide de ses prévisions qu'il augmentera la criticité de l'agent variable le plus critique, ou augmentera suffisamment la

criticité d'une nouvelle variable jusqu'à ce qu'elle dépasse la criticité maximale actuelle de l'agent variable le plus critique. Il se trouve alors en situation de conflit, le résultat de son action détériorant l'état d'un autre agent. Son comportement coopératif l'incite à entreprendre alors la modification de ses plages d'entrée pour ne pas être valide sur la plage de données courante. Ces modifications, réalisées à l'aide des AVT, visent à effectuer la modification minimale sur les plages d'entrée de l'agent contexte dans le sens conduisant à l'exclusion de la situation courante.

SNC2 : Un agent contexte dans un état sélectionné reçoit un message d'un agent variable indiquant que la criticité de la variable n'évolue pas dans le sens décrit par les prévisions de l'agent contexte. Ainsi, au moins une prévision de l'agent contexte n'est pas vérifiée : il se trouve alors en situation d'incompétence.

Cette situation entraîne une modification de la prévision erronée dans le sens de la nouvelle information observée.

SNC3 : Un agent contexte sélectionné reçoit un message d'un agent variable indiquant la modification de la valeur d'une variable du système à contrôler. Cette modification conduit à la sortie des plages d'entrée de l'agent contexte. Ainsi, l'agent contexte qui se considère pertinent, et qui est de plus sélectionné par son agent contrôle associé, en déduit que l'erreur provient de la définition de ses plages d'entrée. De ce fait, il opère une extension de la plage d'entrée en question afin d'englober la situation courante.

SNC4 : Un agent contexte sélectionné reçoit une notification de la part d'un agent contrôle. Celle-ci indique que bien que sélectionné actuellement, l'agent contexte n'est plus le plus pertinent. Ce cas se produit par exemple lorsqu'un nouvel agent contexte se manifeste auprès de l'agent contrôle en offrant de meilleures prévisions. Une situation de conflit apparaît alors. Celle-ci est résolue par la réduction si possible de la durée de l'action de l'agent contexte actuellement sélectionné afin de modéliser le fait qu'il reste pertinent pour la période du début de sa sélection, à l'état courant. De plus, une réduction des plages d'entrée de l'agent contexte est effectuée pour tenter d'exclure la situation courante. Enfin, la réduction de la durée des prévisions de l'agent contexte conformément à la nouvelle durée de son action permet de maintenir la cohérence de l'ensemble des modifications effectuées, en conservant la même échelle de temps pour l'action et les prévisions.

SNC5 :Un agent contrôle reçoit un événement provenant d'un agent variable, marquant une augmentation de la criticité suffisamment importante pour dépasser la criticité maximale des agents variables au moment où l'action à appliquer a été choisie par l'agent contrôle. L'agent contrôle relève alors une situation de conflit. Face à cette situation, l'agent contrôle interrompt

l'action en cours par l'arrêt de l'envoi des modifications à destination des agents variables. De plus, le problème est soumis à l'agent contexte actuellement sélectionné, ce qui génèrera une SNC que cet agent contexte devra résoudre.

SNC6 : Un agent contrôle perçoit une augmentation de la criticité maximale du système, et aucun des agents contextes auxquels il est lié ne se considère comme pertinent pour la situation actuelle. Il se trouve alors en situation d'incompétence. L'agent contrôle tente alors de résoudre cette situation en créant un nouvel agent contexte de la manière suivante :

- Les plages d'entrée du nouvel agent contexte sont générées pour englober la situation courante. Elles utilisent donc les valeurs courantes des variables comme centre d'une plage d'une taille Δ fixée empiriquement.
- Une fois les plages d'entrée initialisées, il reste à définir la forme de l'action à appliquer. Celle-ci est déterminée grâce à l'observation des autres agents contextes présents dans le système, et reliés au même agent contrôle. L'ensemble des actions des agents contextes dont les plages d'entrée sont vérifiées, mais qui ne sont pas valides du fait de leurs prévisions sont étudiées afin de déterminer une nouvelle action pertinente. Par exemple, si l'ensemble de ces actions souhaite diminuer la valeur contrôlée, alors la nouvelle action sera générée afin de l'augmenter. L'idée sous-jacente est que les actions conduisant à une augmentation de la criticité ne sont pas adéquates, et qu'il faut donc initialiser le nouveau contexte à l'aide d'une action sensiblement différente.

L'ensemble de ces comportements compose donc les modules de coopération des différents agents et leur permet ainsi de s'ajuster selon les erreurs qu'ils rencontrent ou de se renforcer lorsqu'ils sont adéquats.

VI.2.3 Bilan

La quantité d'interactions entre les agents dans Obsidian, ainsi que le nombre de situations non coopératives possibles semblent complexifier le fonctionnement de cet AMAS.

Malgré cette complexité apparente, le fonctionnement nominal d'Obsidian est relativement simple. En premier lieu, les agents variables observent les modifications du système à contrôler, et en notifient les agents contextes et contrôles. Les agents contextes recevant ces informations mettent à jour leur état interne et s'ils se considèrent pertinents, se manifestent auprès de l'agent contrôle correspondant. Les agents contrôles reçoivent donc des suggestions argumentées à l'aide de prévisions des actions à appliquer parmi lesquelles ils effectuent leurs choix. Ils transmettront

ensuite ces choix à l'agent variable contrôlable adéquat. Ce dernier agira alors sur le système à contrôler selon l'action définie.

Le contrôle global du système est donc réalisé par des agents contrôles ne possédant aucun lien direct entre eux. Cependant, grâce à l'utilisation des contextes, et en particulier de leurs mécanismes de prévisions, une synchronisation des actions effectuées émerge. En effet, les prévisions d'un agent contexte sont établies à partir des informations qu'il perçoit, et de ce fait elles décrivent les conséquences sur les criticités de l'ensemble des actions appliqués simultanément sur le système à contrôler, et non pas de l'action isolée que cet agent propose. Ainsi, si nous supposons un état d'Obsidian parfaitement adapté au système qu'il doit contrôler, alors les agents contextes sélectionnés à un instant t par l'ensemble des agents contrôles présenteront exactement les mêmes prévisions.

Il est également intéressant de détailler la manière dont Obsidian gère l'existence de plusieurs variables contrôlables : Chacune de ces variables entraîne la création d'un agent variable associé, mais également de l'agent contrôle spécifique à cette variable. Ces agents contrôles possèdent naturellement leur propre ensemble d'agent contextes, agents dont le nombre évolue au cours du temps selon les observations d'Obsidian, les objectifs à atteindre et les actions effectuées.

L'AMAS résultant peut donc être considéré tel que représenté par la figure VI.11, c'est-à-dire comme un ensemble d'AMAS dont chacun est spécialisé dans le contrôle d'une variable unique. Ces différents AMAS sont reliés entre eux par les agents variables. Cette figure présente un système conçu pour appliquer des modifications sur deux variables distinctes, VM1 et VM2. L'intérêt de présenter Obsidian de la sorte réside dans la compréhension de sa gestion du contrôle. En effet, décrire Obsidian comme utilisant un AMAS pour chaque variable contrôlable souligne en premier lieu pourquoi le passage à l'échelle ne pose pas de problème : l'ajout d'une variable contrôlable conduit simplement à l'ajout d'un AMAS de complexité analogue à celle des AMAS préexistant dans le système de contrôle. Ainsi, l'ajout d'un degré de liberté dans les contrôles possibles ne conduit pas à une augmentation exponentielle du nombre d'agents au sein d'Obsidian.

De plus, cette architecture en tant qu'ensemble d'AMAS souligne l'émergence de la synchronisation d'actions simultanées, dans le sens où ces actions sont déterminées sans aucun échange d'information entre les différent agents contrôles ou agents contextes appartenant à des agents contextes distincts. Dès lors, la décision d'Obsidian d'appliquer simultanément plusieurs actions résulte des décisions strictement locales effectuées par les différents AMAS le composant. En cela, la synchronisation des actions de contrôle et la politique générale de contrôle appliquée

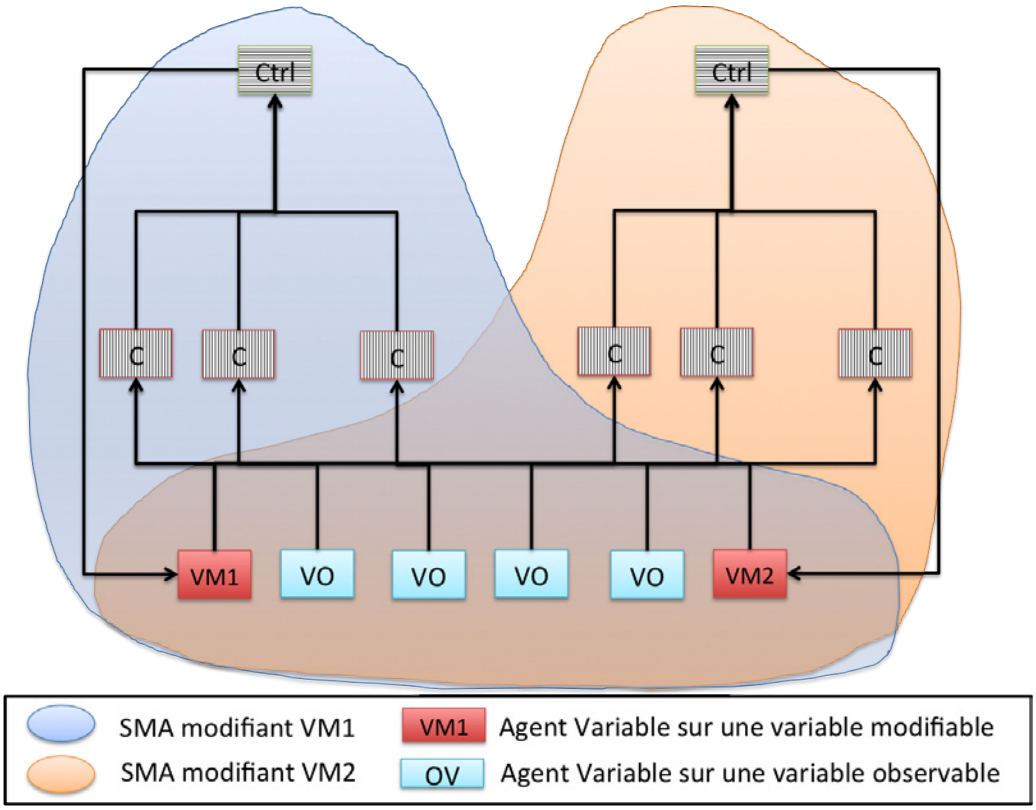

Figure VI.11 – Obsidian en tant que composition d'AMAS

par Obsidian émerge.

Les paragraphes suivants détaillent l'application pratique d'Obsidian pour le contrôle d'un système proies-prédateurs.

VI.3 Application au contrôle d'un système proies-prédateurs

La première application d'Obsidian à la résolution d'un problème pratique concerne le contrôle d'un système proies-prédateurs. Le choix d'un tel système à contrôler a été guidé par la dynamique des populations mise en jeu, dynamique pouvant être observée entre plusieurs micro-organismes au cours de certains bioprocédés par exemple.

Les exemples détaillés dans les paragraphes suivants reposent donc sur le modèle d'un tel système réalisé sous Matlab, contrôlé par une implémentation d'Obsidian réalisée en Java. Obsidian n'a donc accès qu'aux valeurs des variables, et n'a aucune visibilité des calculs effectués par le modèle régissant le système à contrôler.

VI.3.1 Le système proies-prédateurs

Les équations différentielles non-linéaires de Lotka-Volterra [Lotka 1910] forment un modèle dit « proies-prédateurs ». Au-delà de leurs premières applications historiques concernant des espèces animales spécifiques, elles permettent de décrire la dynamique de nombreux systèmes biologiques présentant des interactions entre deux entités, tant au niveau macroscopique que microscopique.

Les deux équations différentielles utilisées pour cette modélisation sont les suivantes :

- $dX/dt = aX - bXY$ (1)
- $dY/dt = cXY - dY$ (2)

Ce système d'équation fait intervenir un certain nombre de variables et de paramètres dont voici la signification :

- X, la quantité de proies ;
- Y, la quantité de prédateurs ;
- a, le taux de reproduction des proies en l'absence de prédateurs ;
- b, le taux de mortalité des proies due aux prédateurs ;
- c, le taux de reproduction des prédateurs en fonction des proies consommées ;
- d, le taux de mortalité des prédateurs.

Lors de l'utilisation classique d'un système proies-prédateurs, les paramètres a, b, c et d des équations (1) et (2) sont définis par l'utilisateur et statiques durant le fonctionnement du système. Cependant, afin de permettre un contrôle du système conduisant les quantités de populations vers des objectifs donnés, il est nécessaire d'y ajouter des points d'entrée, c'est-à-dire des paramètres modifiables dynamiquement en cours de fonctionnement. Cette modification est effectuée en transformant certains des paramètres a, b, c et d en variables que le système de contrôle pourra modifier au cours du temps. Ainsi, le problème revient à atteindre des quantités de populations définies en modifiant les taux de reproduction et de mortalité des proies et des prédateurs par exemple.

Le problème du contrôle des populations de proies et de prédateurs a été décomposé en plusieurs sous-problèmes, de complexité croissante, afin de valider les différents comportements des agents au fur et à mesure du développement. En particulier, la progression des exemples correspond à la vérification et l'implémentation des comportements des agents contextes, s'étendant d'une simulation possédant un nombre fixé et défini d'agents à une simulation capable de créer ces agents contextes lorsque les agents la composant le jugent nécessaire et s'adaptant à des

Tableau VI.1 – Conditions initiales de l'étude préliminaire

Variable	Valeur initiale	Objectif
X	2	5
Y	2	15
a	1	-
b	1	-
c	1	-
d	1	-

modifications dynamiques d'objectifs.

Il est à noter que si les exemples présentés ci-dessous sont définis pour un ensemble de conditions initiales données, des tests semblables ont été réalisés pour des jeux de paramètres différents, ainsi que sur des objectifs distincts avec des résultats similaires.

VI.3.2 Etude préliminaire

Une première étude a été réalisée afin de valider la notion de contexte, et de déterminer la faisabilité du contrôle d'un système proies-prédateurs pour un ensemble de conditions données.

Le problème est ici de contrôler les deux populations (X et Y) vers des objectifs définis, tout en modifiant les deux seules variables a et c. Le choix de ces variables contrôlables a été déterminé expérimentalement afin de rendre le contrôle possible. Les conditions initiales, ainsi que les valeurs objectifs sont définies dans le tableau VI.1.

La dynamique du système proies-prédateurs obéissant à ces conditions initiales et n'étant soumise à aucun contrôle est représentée par la figure VI.12 soulignant clairement le caractère oscillant des populations de proies et de prédateurs.

Dans un premier temps, ce travail a demandé la création d'un modèle algorithmique permettant de représenter la notion de contexte hors du cadre d'un AMAS. Un ensemble de contextes, tels que décrits dans la section VI.2.2.2, a été réalisé à la main de manière empirique afin de former une base de connaissance minimale sur le système à contrôler et d'observer l'application de ces contextes dans un processus de contrôle. Ainsi, le modèle proies-prédateurs faisant office de système à contrôler a été modifié afin d'implémenter à chaque pas de simulation une vérification de l'état de chacune des variables composant le système. Ces variables sont comparées aux plages d'entrée des contextes réalisés à la main, et les actions des contextes se jugeant

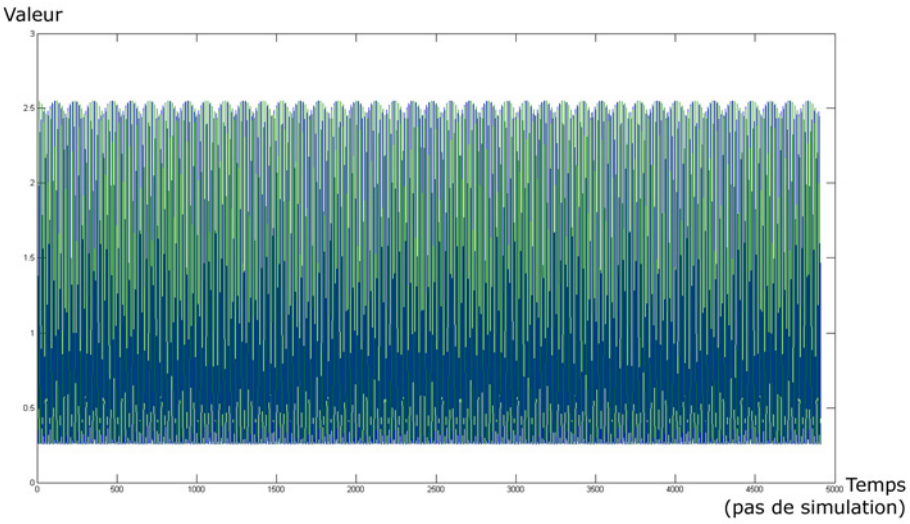

Figure VI.12 – Dynamique du système à contrôler lors d'une absence de contrôle

pertinents sont alors appliquées sur le système à contrôler.

Il est important de noter que les contextes de cette étude préliminaire sont statiques, et de ce fait demeurent identiques aux contextes initiaux conçus à la main tout au long du processus de contrôle. Ces contextes représentent un ensemble de 32 comportements créés de la manière suivante.

Les contextes statiques utilisés sont déterminés par deux facteurs : leurs plages d'entrée et l'action qu'ils proposent. Dans ce problème, deux variables (X et Y) ont un objectif à atteindre et seules deux variables (a et c) sont contrôlables. Les actions appliquées sur le système sont donc l'augmentation ou la diminution de a et c.

Les plages d'entrée des contextes couvrent les différentes combinaisons de cas possibles concernant les valeurs de X et Y par rapport à leurs objectifs (supérieures ou inférieures) en tenant compte de deux vitesses distinctes afin de refléter la différence entre augmentation et diminution de la valeur.

Ainsi, le premier contexte verra ses plages d'entrée de X et Y s'étendre de 0 à leurs objectifs, pour des vitesses positives. Le second utilisera les même bornes mais avec une vitesse positive et une autre négative. Le même processus permet également la définition de plages d'entrée s'étendant de la valeur objectif à une borne haute des valeurs atteignables par les variables. Ce mécanisme est répété jusqu'à l'obtention des 16 combinaisons différentes.

Comme les actions sont possibles sur deux variables distinctes, ces contextes sont donc

dupliqués de manière à composer un ensemble de 16 contextes définissant les actions sur la variable a, et un même ensemble définissant les actions sur c. L'amplitude des actions proposées a été définie empiriquement.

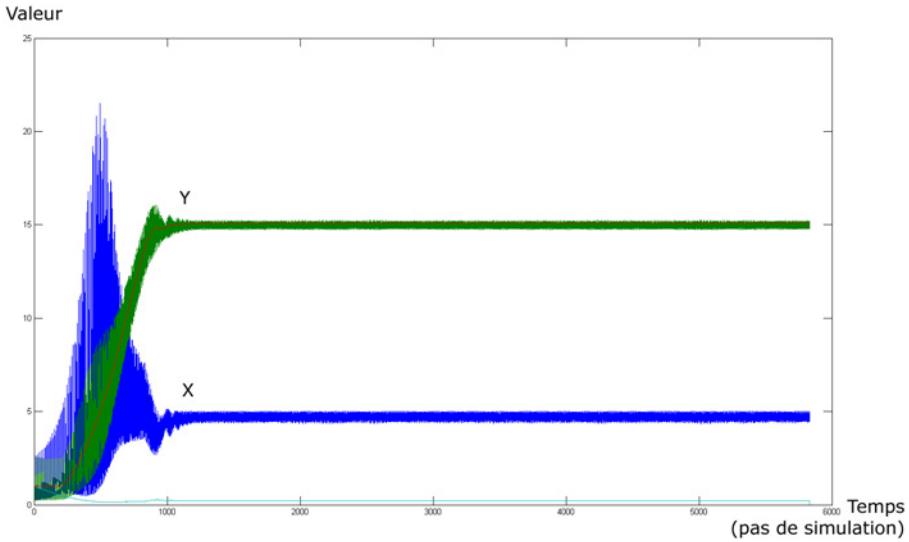

Figure VI.13 – Résultats de l'étude préliminaire (non agentifiée)

La figure VI.13 illustre les résultats obtenus, la courbe verte représentant la quantité de prédateurs, et la courbe bleue la quantité de proies. Le premier point notable réside dans l'atteinte effective des objectifs définis. En effet, les quantités de populations oscillent autour de la valeur attendue par l'utilisateur. Cependant, une hausse importante de la quantité de proies est observable avant que les deux populations ne convergent vers leurs objectifs. Ce phénomène est lié à l'aspect statique des contextes, ne représentant pas de manière suffisamment fine les actions pertinentes à entreprendre afin d'effectuer le contrôle.

Ces résultats soulignent que la notion de contexte permet d'atteindre les objectifs fixés sur ce problème, et note également la nécessité de leur adaptation dynamique afin d'offrir des résultats satisfaisants.

Cet exemple a ensuite été reconduit sur une architecture agent. Les contextes demeurent identiques, statiques, mais cette fois-ci l'ensemble des agents d'Obsidian est instancié, bien que leur comportement coopératif ne soit pas implémenté. Les contextes définis à la main n'étant pas source de SNC, cet exemple permet d'observer la validité de la structure d'Obsidian ainsi que le fonctionnement nominal des agents.

De plus, cette application permet de mettre en œuvre la passerelle entre le système à contrô-

ler s'exécutant sous Matlab et Obsidian codé en Java et s'exécutant dans la console virtuelle.

La figure VI.14 détaille les résultats obtenus. Ceux-ci sont semblables aux résultats de l'implémentation purement algorithmique sous Matlab, Contrairement à cette première implémentation, et du fait de l'agentification, les agents contrôles ne sont plus forcés d'effectuer une action à chaque pas de temps si celle-ci prévoit d'empirer la criticité du système. Ce point explique la diminution de la phase d'augmentation de la quantité de proies en début de simulation, tout comme la convergence légèrement plus lente des deux quantités de populations vers leurs objectifs.

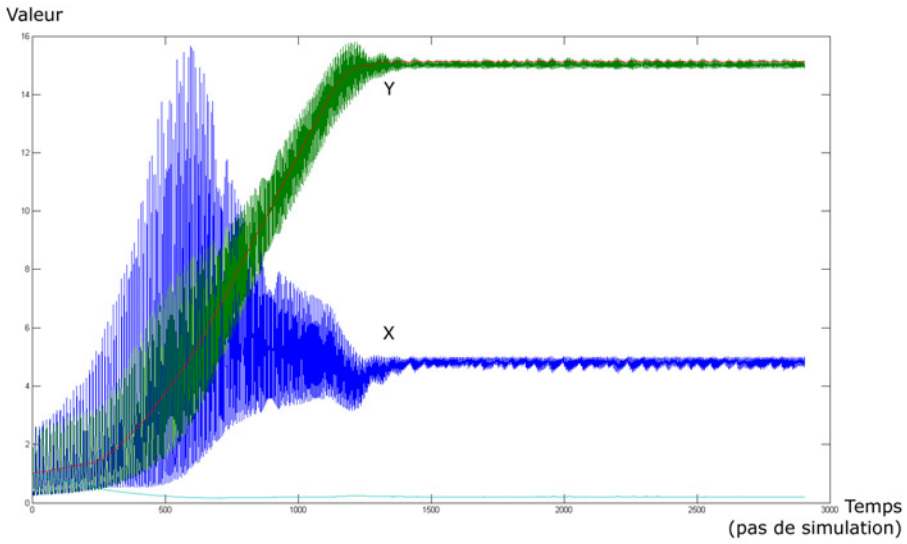

Figure VI.14 – Résultats de la version agentifiée de l'étude préliminaire

VI.3.3 Création dynamique de contextes

Ce second exemple reprend les mêmes équations, ainsi que les mêmes conditions initiales que lors de l'étude préliminaire afin de faciliter leur comparaison.

La différence réside dans le mécanisme de contrôle utilisé qui est cette fois réalisé par une version complète d'Obsidian. De plus, le contrôle doit cette fois-ci être réalisé sans aucun ensemble de contextes initiaux afin de tester les capacités de créations de contextes et par extension, les facultés d'apprentissage d'Obsidian.

La figure VI.15 détaille les résultats obtenus. Ceux-ci laissent apparaitre une nette amélioration des résultats et ce, sur plusieurs points distincts.

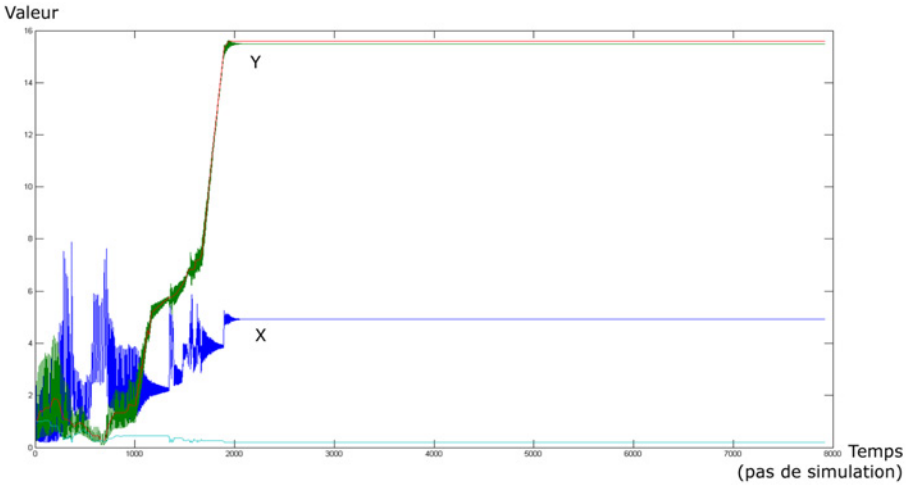

Figure VI.15 – Résultats du contrôle réalisé par Obsidian

En premier lieu, la phase d'apprentissage d'Obsidian, visible de $t = 0$ à $t = 1500$ présente malgré son aspect chaotique caractéristique une atténuation significative du phénomène d'explosion de la quantité de proies observable sur les deux exemples précédents.

Dans un second temps, la stabilité des résultats une fois leur objectif atteint (avec une erreur inférieure à 1.5 pourcents) permet de souligner la cohérence du comportement des agents et en particulier de l'utilisation de la criticité : le système se trouve dans un état coopératif et les agents sont satisfaits, leurs actions maintiennent donc cet état. La figure VI.16 illustre cette diminution progressive de la criticité des variables X et Y au long de la simulation.

Enfin, la vitesse de convergence vers les objectifs, plus lente que lors de l'implémentation précédente, doit ce ralentissement à la phase d'apprentissage initiale se déroulant en même temps que le contrôle. Les exemples précédents utilisaient un ensemble de contextes prédéfinis et suffisant pour effectuer cette tâche alors qu'Obsidian ne possède aucun contexte lors du lancement de la simulation. Il est également remarquable qu'un ensemble final de 26 contextes soit défini par Obsidian, soulignant son apprentissage du contrôle vers un objectif précis, et non une définition générale des actions à entreprendre pour des cas auxquels il n'est pas confronté. Le nombre de contextes obtenu au final varie selon les simulations, mais reste compris entre 22 et 28 pour ce problème précis. Les contextes obtenus pour cet exemple en particulier sont représentés par les figures VI.17 et VI.18. Ces deux figures se complètent, c'est-à-dire que le premier contexte présent dans Obsidian aura comme plage d'entrée sur X la plage représentée par la première colonne de la figure VI.17, et comme plage d'entrée sur Y la plage représentée par la première colonne de la figure VI.18. Les plages d'entrée des 26 contextes sont donc

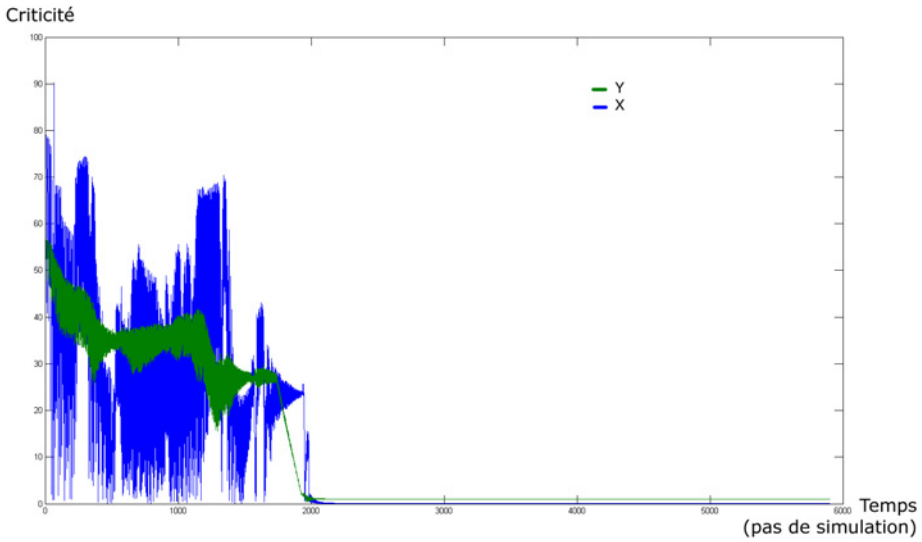

Figure VI.16 – Evolution des criticités de X et Y au cours du contrôle par Obsidian

fournies par la combinaison de ces deux représentations et la couleur de chacune des colonnes fournit une indication sur la vitesse associée à la plage d'entrée, le rouge représentant une vitesse positive tandis que le bleu symbolise une vitesse négative. Enfin, il est à noter que les contextes semblant représenter la même situation, tels que les contextes 23 et 24 correspondent en pratique à des actions associées sur des variables différentes.

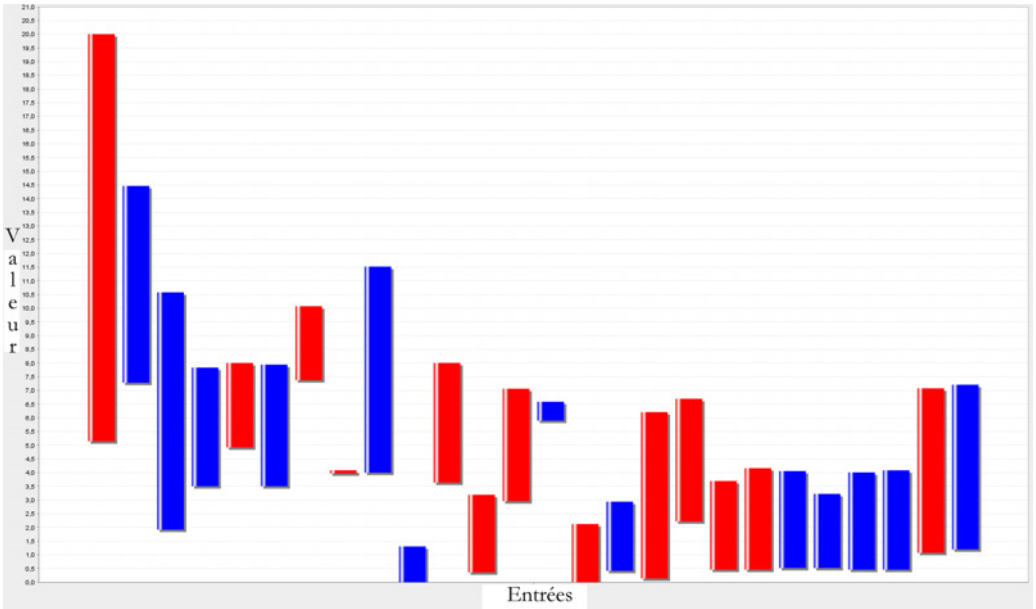

Figure VI.17 – Répartition des plages d'entrée des contextes sur X

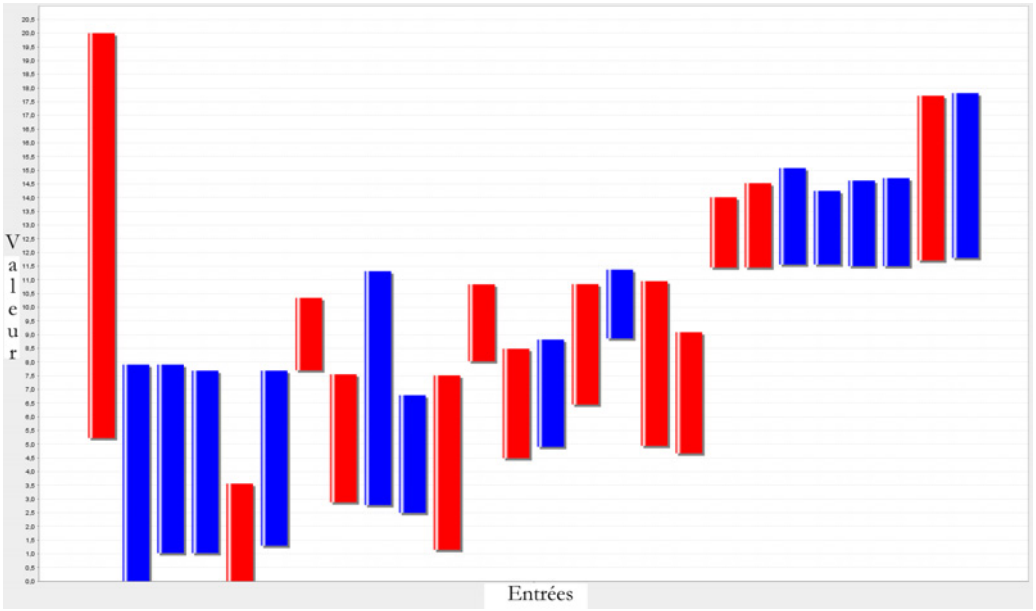

Figure VI.18 – Répartition des plages d'entrée des contextes sur Y

Tableau VI.2 – Conditions initiales de l'étude préliminaire

Variable	Objectif
X	5
Y	15, 13, 8

Obsidian est donc en mesure d'apprendre à contrôler ce système sans contexte a priori et ce, tout en augmentant la qualité des résultats obtenus par rapport à un ensemble de contextes créés à la main. Ainsi, sa capacité d'apprentissage lui permet de conduire un procédé inconnu vers un objectif défini.

VI.3.4 Modifications dynamiques des objectifs

Ce dernier exemple utilise également la version d'Obsidian incluant l'ensemble des comportements des agents. Cette fois-ci, le système à contrôler part des mêmes conditions initiales, mais les objectifs définis sur la variable Y sont modifiés en cours de simulation. Ainsi, les capacités d'adaptation d'Obsidian sont testées par sa réactivité et la qualité de sa réponse lors d'un changement d'objectif. L'ensemble des objectifs défini dans le tableau VI.2 met en évidence les trois objectifs distincts que la variable Y devra atteindre au cours de la simulation.

La figure VI.19 illustre le résultat de cette application. La réaction d'Obsidian aux modifi-

cations des objectifs est immédiate, et sa convergence vers un nouvel objectif possible malgré l'utilisation d'un ensemble de contextes ayant été créé pour un objectif différent.

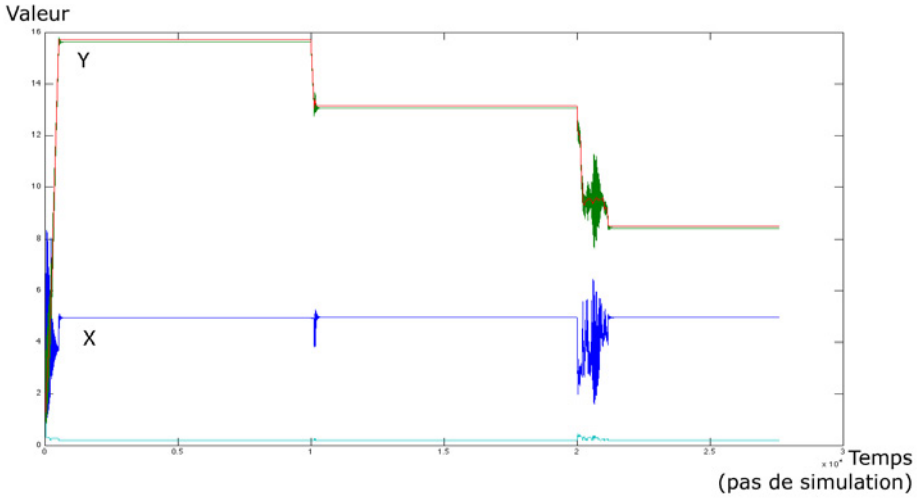

Figure VI.19 – Résultats du contrôle réalisé par Obsidian pour des objectifs dynamiques

En effet, Obsidian ne possède initialement aucun contexte, il constitue donc un ensemble de contextes lui permettant d'atteindre son premier objectif. Une fois l'objectif modifié, l'ensemble des contextes existant alors forme une base de connaissances potentiellement erronées, le nouvel objectif ne pouvant pas forcément être atteint par leur utilisation. La capacité d'Obsidian à conduire un procédé vers les nouveaux objectifs souligne donc l'efficacité de ses comportements menant à la révision de ses connaissances erronées, autrement dit, sa faculté de gestion des contextes. Ce point est en particulier visible sur les figures VI.20 et VI.21 faisant intervenir une quantité finale d'agents contextes inférieure à celle de l'exemple précédent malgré une complexité du problème accrue. En effet, les contextes obsolètes sont modifiés et fusionnés afin de rendre compte des objectifs courants à atteindre, les contextes ayant été utilisés pour conduire le système vers le premier objectif n'existant plus dans l'ensemble final des contextes.

Enfin, la figure VI.22 met en évidence l'évolution du calcul des criticités selon les modifications des objectifs. Lorsqu'un objectif est défini, sa criticité connaît une forte augmentation, et se retrouve peu à peu réduite jusqu'à la satisfaction de l'agent au fur et à mesure qu'Obsidian apprend à contrôler le système.

Cet exemple souligne les capacités d'adaptation dont fait preuve Obsidian lorsqu'il se retrouve confronté à des modifications en cours de fonctionnement. Ses capacités de révision des connaissances sont également mises en avant, lui permettant ainsi d'optimiser la quantité de d'agents contextes utilisés.

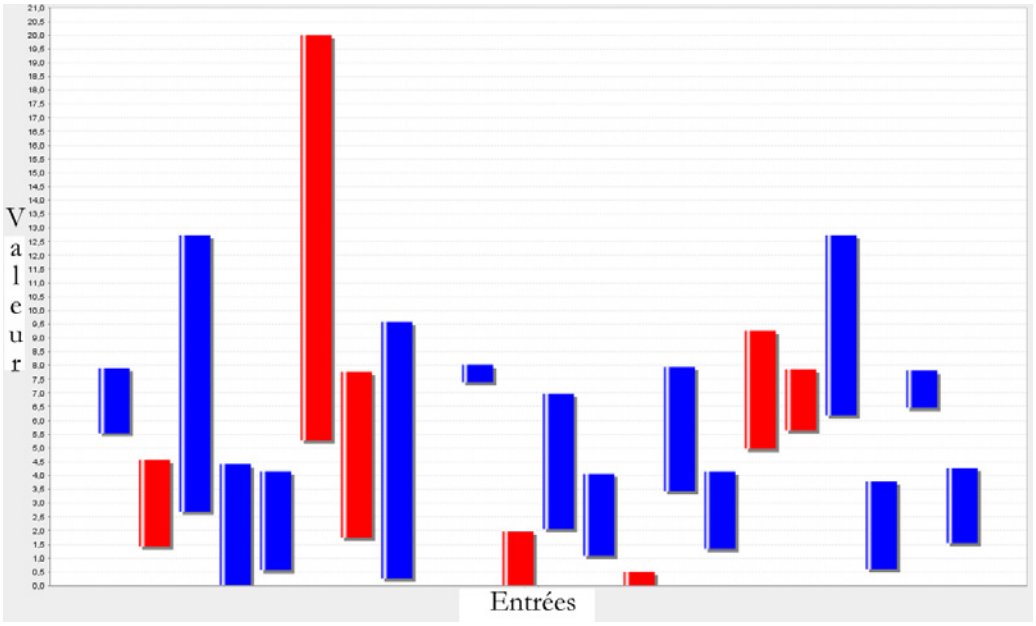

Figure VI.20 – Répartitions des plages d'entrée des contextes sur X

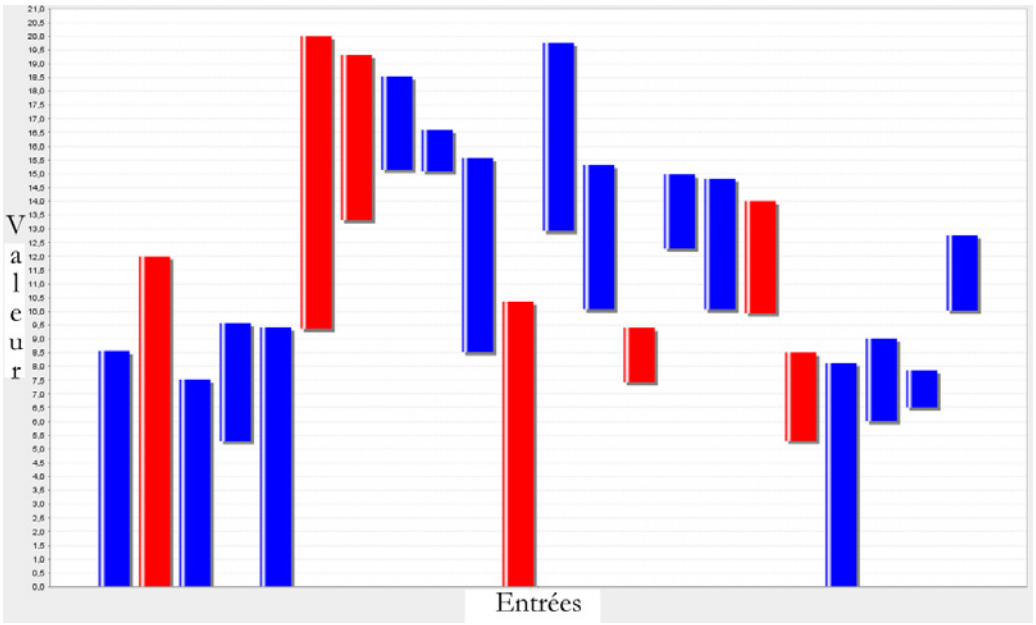

Figure VI.21 – Répartitions des plages d'entrée des contextes sur Y

VI.4 Application au contrôle de bioprocédés

L'application finale vers laquelle tend Obsidian réside dans le contrôle d'un bioprocédé réel. Avant de réaliser cette étape, il est nécessaire de tester le système de contrôle dans des conditions les plus proches possibles de la réalité. Ce test est rendu possible par le contrôle d'un système

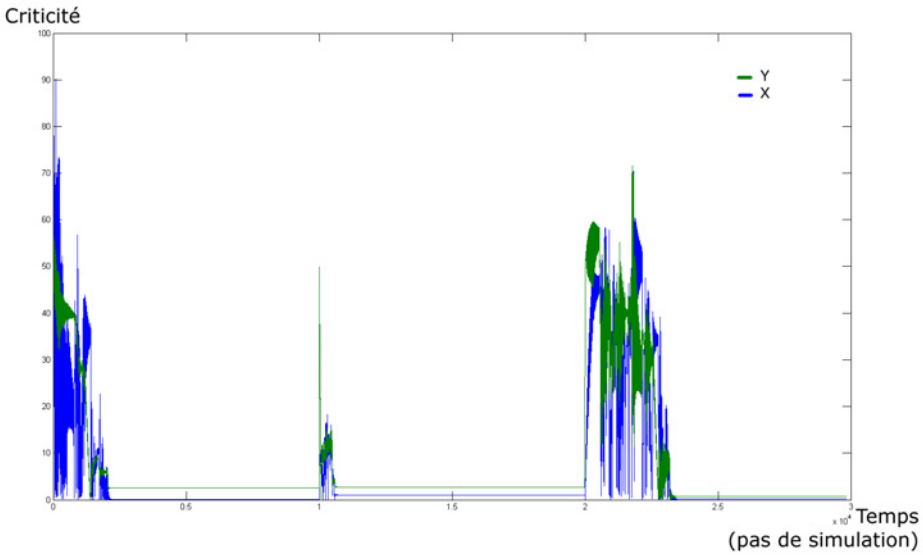

Figure VI.22 – Evolution des criticités de X et Y au cours du contrôle par Obsidian

informatique décrivant le comportement d'un bioprocédé.

Les paragraphes suivants détaillent la mise en œuvre d'un tel système à contrôler, son fonctionnement ainsi que l'application d'Obsidian pour assurer son contrôle.

VI.4.1 Le système à contrôler

Le système cible utilisé modélise le fonctionnement d'un bioprocédé en spécifiant de manière distincte la physique du bioréacteur et l'évolution des micro-organismes situés à l'intérieur. L'objectif de ce système n'est pas de modéliser un bioprocédé précis, mais de fournir une image réaliste de l'évolution d'une population de micro-organismes décrite par le modèle biologique selon l'ensemble des conditions régnant dans le fermenteur.

Cette description est centrée sur l'utilisation d'un ensemble d'équations mathématiques représentant les lois physiques régissant les interactions entre les différents éléments présents. Ce sont au total 32 équations différentielles manipulant un ensemble de 103 paramètres qui interviennent afin de modéliser à la fois la partie fermenteur et les paramètres biologiques.

VI.4.1.1 Le modèle biologique

Le modèle biologique est une simplification du métabolisme d'*Escherichia coli* et ne considère que son métabolisme oxydatif sur glucose, laissant de côté le métabolisme oxydo-réductif

correspondant aux situations pour lesquelles le transfert d'oxygène est limitant. Dans ces conditions, le modèle utilisé ne fait apparaître que la surproduction d'acide acétique.

Ce modèle biologique est donc un modèle phénoménologique décrivant la croissance des micro-organismes, la production et la consommation d'acide acétique en fonction des variables d'état et des conditions opératoires de température et pH, tout en considérant l'effet inhibiteur de l'acide acétique.

Ainsi, les micro-organismes seront soumis à l'ensemble des réactions suivantes selon l'état du système :

- Réaction de croissance, décrivant l'évolution de la quantité de micro-organismes selon les quantités d'azote, d'oxygène, de glucose et d'acide acétique.
- Réaction de production d'acide acétique, liée à la réaction de croissance qu'il est nécessaire de limiter au maximum à cause du caractère inhibiteur de l'acide acétique.
- Réaction de consommation d'acide acétique, observée lorsque la concentration de glucose est très limitante, mais inhibée lorsque la concentration en acide est trop importante.
- Réaction de consommation du glucose, dépendant principalement de la concentration en substrat et en oxygène dissous.
- Réaction de consommation d'azote pour la croissance, couplée à la réaction de croissance.
- Réaction de consommation d'oxygène.
- Réaction de production de CO_2.

L'objectif des contrôles à réaliser par Obsidian va donc concerner l'augmentation de la quantité de micro-organismes tout en maintenant la quantité d'acide acétique au minimum en évitant sa création ou en le consommant. Ainsi, la croissance des micro-organismes est optimisée par la minimisation de l'effet inhibiteur de l'acide acétique. La difficulté de cette tâche provient du lien existant entre la croissance de ces micro-organismes et la production d'acide acétique impliquant de conduire cette croissance en tenant compte de ce phénomène inhibiteur.

VI.4.1.2 Le modèle du fermenteur

Ces micro-organismes sont plongés dans un environnement sur lequel l'opérateur, et par extension Obsidian, peut agir. La figure VI.23 résume les principales interactions entre les variables physiques de cet environnement et souligne donc les répercussions potentielles que la modification de l'une d'entre elles produit sur les autres et, par extension, son impact éventuel sur les micro-organismes.

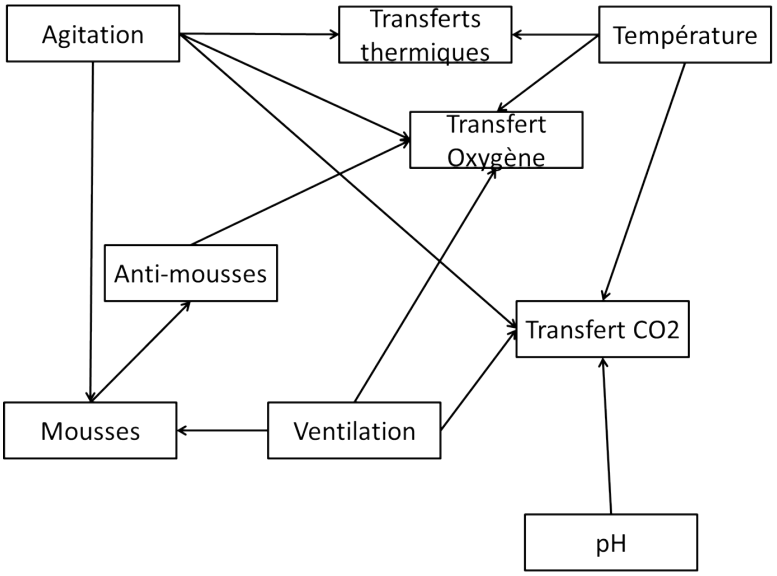

Figure VI.23 – Interactions physiques au sein du fermenteur

Ainsi, ce point souligne la complexité des contrôles à opérer car chacune des actions entreprises sur l'environnement produit un ensemble de conséquences observables à plusieurs niveaux, physique ou biologique. Ainsi, une simple modification telle que la variation de la température va, entre autres, modifier les transferts gazeux et influer différemment sur les différentes réactions auxquelles sont soumis les micro-organismes.

L'algorithme de simulation du bioprocédé, réalisé sous Matlab [1], utilise donc la résolution de ces équations différentielles décrivant l'ensemble des modèles afin de calculer l'évolution du système et de fournir une image pertinente de l'évolution d'un tel procédé. Cette image demeure cependant une version simplifiée d'un bioprocédé réel, certains phénomènes tels que les flux de sels ou les variations de volumes liées aux différents débits n'étant pas pris en compte, mais celle-ci reste très réaliste pour l'ensemble des hypothèses mentionnées.

Il est important de préciser que l'ensemble des paramètres intervenant dans ces équations n'est pas observable par Obsidian, ce dernier n'ayant accès qu'aux variables réellement mesurables *en ligne*, soit moins d'une vingtaine selon le scénario étudié. Cependant, l'ensemble des paramètres peut être modifié par l'utilisateur pour créer des scénarios de contrôle distincts obéissant à des contraintes précises.

1. Un extrait du modèle réalisé sous Matlab se trouve en Annexe 2

VI.4.2 Application d'Obsidian

Appliquer Obsidian à un système aussi complexe n'est pas foncièrement différent du travail réalisé dans les exemples précédents. En effet, de par la conception même d'Obsidian, la seule différence réside dans le nombre de variables plus élevé dans cet exemple-ci. Le fonctionnement interne du système à contrôler n'a, quant à lui, pas besoin d'être connu et sa complexité n'influence donc pas la phase d'application.

VI.4.2.1 Définition des criticités

L'étape la plus complexe de l'application d'Obsidian réside dans la définition des criticités des différentes variables présentes dans le bioprocédé à contrôler. En effet, cette définition nécessite un certain nombre de connaissances générales sur les caractéristiques physiques des éléments manipulés.

La définition de ces criticités doit donc être réalisée par un expert du domaine en tenant compte des objectifs à atteindre. Considérons, par exemple, que nous souhaitions connaître la criticité de la température dans un scénario visant à augmenter la production de micro-organismes. La définition de la criticité de la température dépend donc de son influence sur la croissance de ces micro-organismes.

La première étape consiste à modéliser cette influence représentée par une fonction ($f_{(\theta)}$). Cette fonction doit tenir compte de deux aspects :

- La loi d'Arrhénius, décrivant la variation de la vitesse de la réaction chimique étudiée et fournissant l'équation suivante : $\mu = \mu^c_\infty \cdot e^{-\frac{E_a}{R \cdot T}}$
- Le phénomène d'inactivation apparaissant au-delà de la température idéale de 35.7°C et annulant la croissance pour des températures inférieures à 0°C ou supérieures à 45°C.

De ce fait, la formule générale décrivant l'influence de la température sur la cinétique de croissance des micro-organismes est la suivante : $f_{(\theta)} = \mu^c_\infty \cdot e^{-\frac{E'_a}{T}} \cdot \frac{(\theta_{\max}-\theta)^3}{K^i_\theta+(\theta_{\max}-\theta)^3}$

La courbe du calcul de la criticité correspondant à ce phénomène est alors obtenue en associant la criticité minimale (la meilleure) au taux de croissance maximum, ce qui produit le résultat représenté par la figure VI.24.

L'expert du domaine définit donc les criticités des variables observables - telles que le pH, la concentration d'acétate ou encore la concentration en CO_2 - de manière similaire afin de fournir aux agents variables d'Obsidian un lien entre leur satisfaction propre et la réalité biologique.

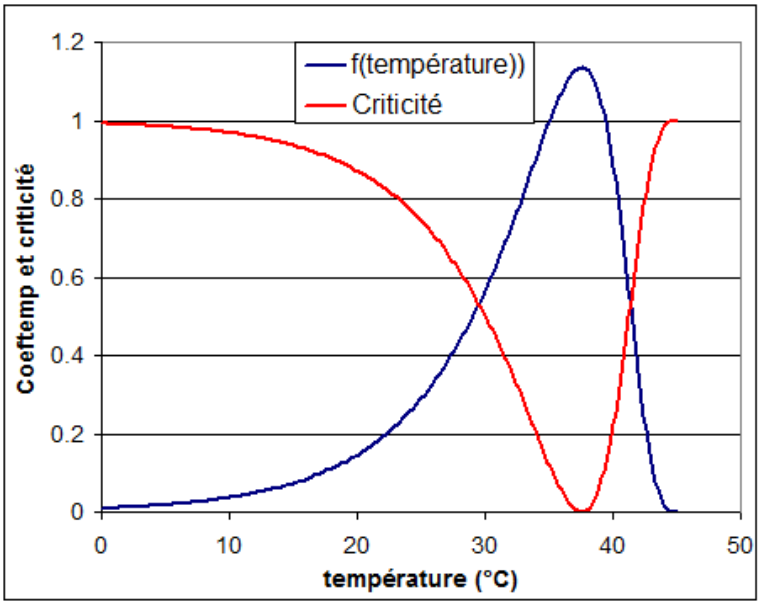

Figure VI.24 – Impact de la température sur la cinétique des micro-organismes et criticité

VI.4.2.2 Définition des objectifs

La seconde étape propre au contrôle de bioprocédés réside dans la vérification de la pertinence biologique de la définition des objectifs et des points d'entrée sur le système à contrôler.

Ceux-ci permettent alors, en complément au paramétrage effectué par l'utilisateur, de déterminer un ensemble de scénarios de test de difficulté croissante, en permettant, par exemple, de n'agir que sur un nombre limité de variables.

La définition des objectifs peut faire appel à une étape supplémentaire de modélisation. Si, par exemple, nous souhaitons maintenir le taux de croissance des micro-organismes à son niveau maximum, l'ajout de cette notion de taux de croissance en tant qu'agent variable doit alors être effectué car il ne s'agit pas d'une variable directement observable sur le bioprocédé. Il est donc nécessaire de créer l'agent représentant cette notion afin d'y associer la criticité correspondant à son objectif. De plus, l'utilisateur doit également fournir un modèle intermédiaire permettant de calculer cette valeur afin de la relier à l'évolution des variables observables et d'être en mesure de mettre sa criticité à jour.

Concernant l'application des contrôles sur ce système, il existe deux types de points d'entrée utilisables :

– Tout d'abord, la modification directe de la valeur d'une variable par Obsidian représente le mécanisme d'action le plus évident. Il sera alors question, par exemple, de modifier

l'apport en substrat afin d'atteindre un objectif donné.
- La seconde approche consiste à charger Obsidian de la modification, non pas des variables directement, mais de l'ajustement dynamique des paramètres des mécanismes de contrôle déjà en place sur le bioprocédé. Ceci équivaut, par exemple, à l'adaptation dynamique de la configuration d'un régulateur PID.

Dans ces deux cas, l'amplitude de modification maximale doit donc être définie selon le scénario à contrôler, afin de conserver la pertinence biologique souhaitée et d'éviter les tentatives d'application de modifications dont les valeurs feraient diverger le système à contrôler.

L'application d'Obsidian au contrôle de ce système nécessite donc la définition d'un scénario décrivant les points suivants :

- Un ensemble précis de conditions initiales représentant l'état des paramètres du système à contrôler. Ces conditions initiales détaillent l'état du fermenteur.
- La définition de la criticité des variables impliquées.
- La définition d'un (ou de plusieurs) objectif(s) à atteindre, accompagnée éventuellement de la création des modèles secondaires nécessaires à leur évaluation.
- La définition des points d'entrée sur le système, c'est-à-dire les variables sur lesquelles Obsidian peut agir et dans quelle mesure.

VI.4.3 Résultats

Le scénario testé ici représente le contrôle de la température du moût de fermentation par Obsidian. Ce contrôle est initialement assuré au sein du modèle biologique par un régulateur PID. La première phase du test réside donc dans l'obtention de résultats témoins obtenus à l'aide de ce régulateur. Dans un second temps, ce dernier est supprimé afin de laisser l'évolution de la température guidée par Obsidian.

Le choix de ce scénario permet de confronter les résultats obtenus avec ceux issus du contrôle opéré par un régulateur PID tout en soulignant une différence essentielle : le régulateur PID connaît a priori la température à laquelle doit être maintenu le moût de fermentation (37 degrés dans le cas présent) alors qu'Obsidian doit la déterminer en cours de fonctionnement.

Afin de réaliser ce contrôle, Obsidian observe l'évolution d'un certain nombre de variables sur lesquelles il possède des criticités définies à partir de connaissances biologiques générales. Le choix de ces informations comme signaux utilisés par Obsidian provient d'une part de leur disponibilité, ce sont en effet des mesures directement réalisables sur le bioprocédé ou estimables

à l'aide de calculs simples, ainsi que de l'existence de fonctions estimant la criticité de chacune d'entre elles. Ces fonctions ont été établies de manière similaire à la description réalisée dans la partie VI.4.2.1. Il est intéressant de noter que ces variables informent sur l'évolution des micro-organismes composant le système à contrôler et ne concernent pas directement l'évolution attendue de la variable contrôlable

Les variables sur lesquelles Obsidian possède des criticités sont les suivantes :

- Le pH, dont la plage de valeur de variation maximale gérée par le modèle biologique est comprise entre 4,7 et 9. L'influence du pH sur la croissance du micro-organisme est liée aux mécanismes énergétiques de ce dernier, induits par la différence de pH intra et exo-cellulaire. Ce mécanisme peut être représenté à l'aide de la composition de deux fonctions sigmoïdales et, par extension, conduire à la définition d'une fonction de criticité maximale pour les valeurs des bornes de définition, et minimale pour des valeurs proches de 6,4.
- Le rapport N/C qui représente le rapport entre l'azote fourni et le carbone accumulé. Ce rapport doit être compris entre 0,1 et 0,6, pour une valeur optimale de 0,23. Ces données permettent donc d'établir une fonction de criticité en tant que combinaisons de fonctions linéaires associant une criticité maximale aux valeurs des bornes (0,1 et 0,6) et nulle pour la valeur attendue (0,23). Il est à noter que ce rapport est calculé à partir de données qui ne peuvent pas être mesurées directement sur le bioprocédé. La nécessité de les calculer impose donc de disposer d'une quantité d'information plus importante, diminuant la pertinence initiale de cet indicateur, et le rendant particulièrement sensible au bruit. Ainsi, la criticité de ce rapport N/C peut être considérée comme pertinente à l'issue de plusieurs pas de simulations, telle que le détaillera la suite du problème.
- Le rendement Carbone qui représente le rapport entre le carbone accumulé et le carbone fourni. L'objectif au cours du contrôle du procédé est le maintien de ce rendement à une valeur de 0,55.
- La quantité de CO_2 (mole/litre), dont une quantité élevée peut inhiber la croissance des micro-organismes. Elle doit donc être maintenue à un niveau faible mais non nul.
- La concentration massique en acide acétique dont la présence inhibe la croissance des micro-organismes. Bien que cet acide puisse être consommé dans certaines situations, il est nécessaire de limiter au maximum sa production.

L'évolution de la température conduite par le régulateur PID intégré au modèle du bioprocédé est représentée par la figure VI.25 soulignant son maintien autour de 37 degrés et servant de témoin à notre expérience.

Une fois ces résultats témoins obtenus, une nouvelle simulation est lancée selon les mêmes

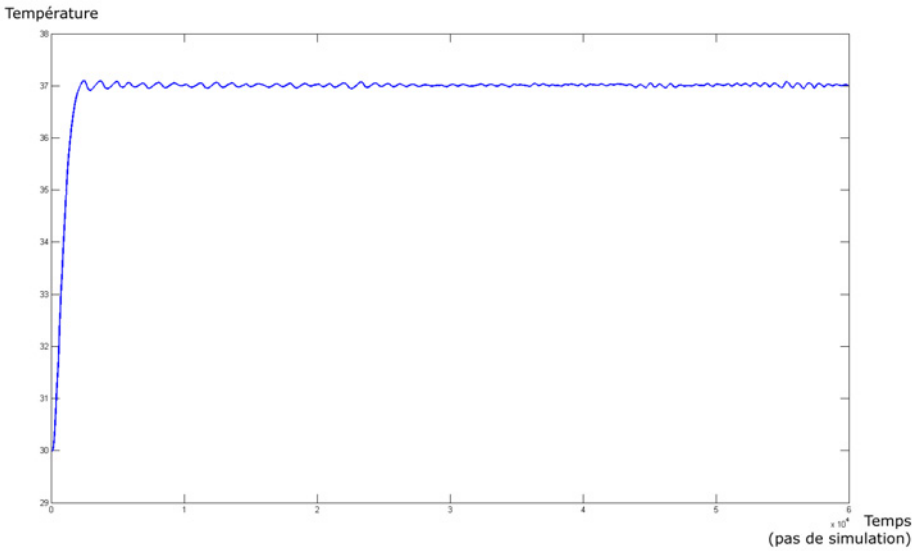

Figure VI.25 – Contrôle de la température effectué par un régulateur PID

paramètres tout en désactivant le régulateur PID présent sur la température pour laisser son contrôle à la charge d'Obsidian. Le logiciel prend en pratique la suite du contrôle effectué par le PID au bout d'un certain nombre de pas de simulation nécessaire à l'obtention de criticités pertinentes. Ce délai correspond à un temps $t = 500$ sur l'échelle présente en abscisse des résultats.

La figure VI.26 représente le contrôle effectué par Obsidian sur la valeur de la température.

De $t = 500$ (la prise en main du système à contrôler par Obsidian) à $t = 1000$, une diminution de la température peut être observée. Ce phénomène correspond à une phase d'ajustement des contextes d'Obsidian alors que les criticités des valeurs observées (en particulier celle du rapport N/C) sont fortement perturbées, et donc de pertinence insuffisante. Les données sur lesquelles le raisonnement des agents est basé conduisent donc temporairement le système vers une solution relativement sous-optimale.

Cependant, dès que ces criticités deviennent suffisamment pertinentes, c'est-à-dire qu'elles sont calculées à partir d'un ensemble de valeurs significatives, Obsidian ajuste l'action effectuée et converge vers la valeur de 37 degrés autour de laquelle il oscille. On retrouve alors les résultats obtenus à l'aide du régulateur PID intégré au modèle biologique, à la différence qu'Obsidian ne possède aucune information sur la température cible, et la détermine selon l'évolution du système à contrôler. Ce test souligne également la capacité d'intégration d'Obsidian à un système possédant déjà un ensemble de contrôles, et sa faculté à agir en remplacement de l'un

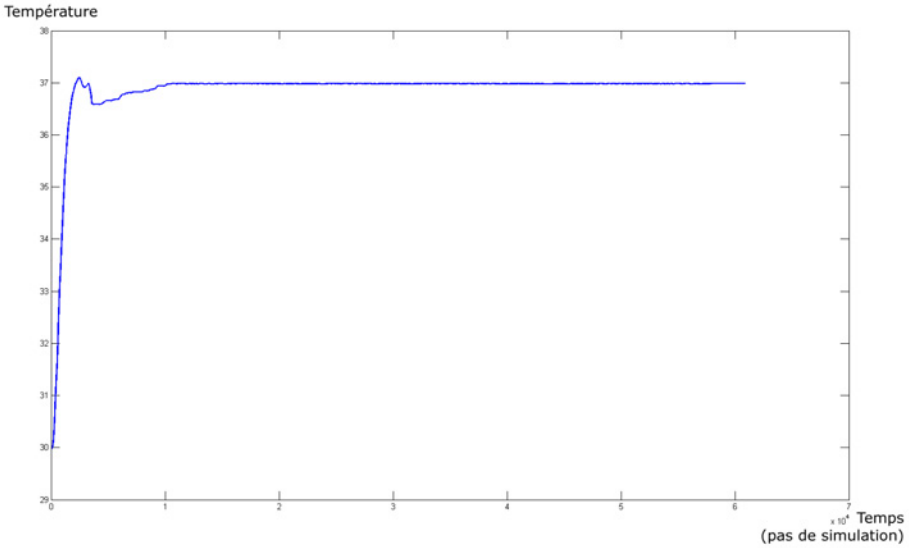

Figure VI.26 – Contrôle de la température effectué par Obsidian

d'entre eux ou lors de certaines situations spécifiques définies par l'utilisateur. Ce dernier peut en effet limiter les actions réalisées par Obsidian à certains pas de temps particuliers, offrant ainsi la possibilité à Obsidian d'apprendre et de former une base minimale de contextes à partir de données pertinentes, réduisant ainsi les erreurs d'apprentissage réalisées lors de la totale découverte d'un nouveau système à contrôler.

L'étude de l'évolution des criticités représentées par la figure VI.27 renforce ces résultats. Les premières valeurs des criticités sont fortement bruitées et sont responsables du phénomène de diminution de température observé en début de contrôle. Malgré cela, les actions entreprises par Obsidian conduisent à une diminution globale de la criticité des différentes variables observées, soulignant par là même la pertinence des actions effectuées par rapport à ces informations.

Ainsi, à partir d'informations biologiques générales, Obsidian arrive à déterminer en temps réel la température du moût de fermentation satisfaisant au mieux le système à contrôler et ce, malgré les fortes perturbations rencontrées lors des mesures des criticités utilisées et que contrairement au régulateur PID, Obsidian ne dispose d'aucune consigne sur la température. Il est intéressant de noter que ces informations utilisées ne concernent pas directement la variable manipulée, le mécanisme de contextualisation d'Obsidian se chargeant de déterminer les liens entre état du système et action à appliquer.

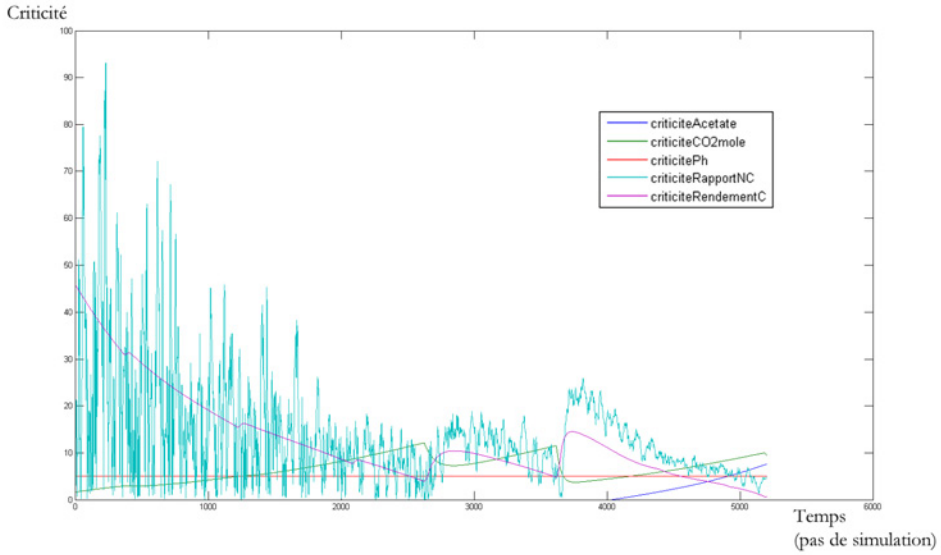

Figure VI.27 – Evolution de la criticité des variables observées par Obsidian

VI.4.4 Généricité de l'approche

L'application d'Obsidian aux proies-prédateurs portait sur le contrôle de toutes les variables autorisées. Dans le bioprocédé, une seule variable était contrôlée par Obsidian, les autres l'étant par des PID. On pourrait penser que les résultats ne sont que partiels et difficilement généralisables si Obsidian contrôlait toutes les variables. Ce n'est pas du tout le cas, car ce sont des systèmes multi-agents auto-adaptatifs qui contrôlent chacune des variables de manière entièrement autonome. Ainsi, remplacer chaque contrôleur par un SMA n'influe aucunement la capacité d'adaptation des autres (capacité d'adaptation dont on a pu juger l'efficacité pour la température). Ce sont uniquement des considérations techniques qui rendent difficiles le remplacement complet des PID dans le simulateur du bioprocédé.

VI.5 Bilan

Ce chapitre a donc détaillé Obsidian, un AMAS capable de contrôler des systèmes complexes sans nécessiter l'utilisation de modèles décrivant le système à contrôler. Son application à des problèmes de difficulté croissante a permis de valider cette approche, tout en illustrant l'impact pratique des différents choix de conception détaillés dans la partie VI.2.

Le système développé s'illustre particulièrement du fait de ses capacités d'apprentissage et d'adaptation, conférant à Obsidian une très grande généricité et permettant ainsi son utilisation

sur un large panel de problèmes, et ce, quel que soit leur domaine d'application.

De ce fait, ce système ne souffre pas de la limite énoncée lors du bilan du chapitre V de par l'absence de modèle du système à contrôler comme base du contrôle. Les agents contextes constituent une forme de modèle, mais un modèle de la politique de contrôle à appliquer afin d'atteindre un objectif précis et non un modèle du système à contrôler. En cela, l'existence du second type de contrôle présenté dans le chapitre IV.2.6 est confirmée de par son utilisation comme base d'Obsidian.

Les résultats obtenus soulignent également la grande généricité existant dans les données nécessaires au fonctionnement du système. Que ce soit avec un ensemble de contextes initial ou sans celui-ci, Obsidian est en mesure d'agir selon les objectifs de l'utilisateur en affinant ou en créant les connaissances lui permettant d'accomplir cette tâche.

La principale limite de cette approche provient de l'implémentation actuelle de la fusion des contextes. En effet, celle-ci bien que très efficace pour permettre l'adaptation d'Obsidian, se révèle trop systématique et entraîne un oubli des situations passées lorsque le système s'est totalement adapté à une nouvelle dynamique. Par exemple, le cas présenté lors d'un changement d'objectif illustre ce phénomène. Ainsi, les contextes ne constituent pas une mémoire à long terme des politiques de contrôle applicables, et un mécanisme de confiance en la pertinence d'un contexte peut améliorer les résultats en prévenant la suppression de contextes pertinents pour une situation donnée, situation dans laquelle le système à contrôler ne se trouve plus.

Ceci permettrait un apprentissage du contrôle à partir de l'exécution de plusieurs simulations successives, se basant chacune sur les contextes issus de la fin de l'itération précédente. Ainsi, par raffinages successifs, la précision et la qualité globale des contextes obtenus sera améliorée. La mise en place de cette fonctionnalité constitue donc la prochaine étape à franchir dans la réalisation d'Obsidian.

Chapitre VII

Conclusion générale et perspectives

Cette thèse se situe à la frontière de plusieurs domaines scientifiques distincts tels que l'informatique, la biologie ou encore l'automatique. Le travail réalisé a donc parcouru ces différents domaines et dressé leurs états de l'art afin d'en extraire une problématique précise : comment concevoir un système de contrôle générique ne souffrant pas des limites des solutions de contrôle actuelles ?

Cette question nous a conduits à l'étude des systèmes multi-agents et en particulier des AMAS qui proposent une approche émergente dont les caractéristiques satisfont les contraintes de notre problème. L'utilisation de cette technologie, encore très faiblement employée dans le cadre précis du contrôle, a finalement abouti à deux propositions distinctes décrivant des systèmes de contrôle se basant sur un ensemble d'hypothèses différent.

Les solutions mises en œuvre reposent toutes deux sur l'utilisation de l'émergence afin de produire le contrôle attendu mais donnent lieu au développement de deux architectures logicielles au fonctionnement bien distinct.

La première solution se base sur la présence de modèles, peu importe leur type, pour guider son raisonnement tandis que le seconde doit générer une politique de contrôle pertinente uniquement à l'aide de l'observation de l'évolution de certaines variables.

Ces deux solutions ont chacune fait l'objet d'une implémentation, conduisant ainsi à la création de deux systèmes informatiques, Malachite et Obsidian, validés à l'aide de leur application à la résolution de divers problèmes.

Le développement de ces systèmes s'est accompagné de nombreux enseignements, concernant tant les problématiques scientifiques auxquelles ils se confrontaient directement que la méthodologie de recherche en général et les spécificités liées à la pluridisciplinarité du sujet.

VII.1 Les apports informatiques

Les apports de cette thèse, d'un point de vue informatique, sont visibles à plusieurs niveaux, et la plupart des objectifs définis en tant que problématique scientifique, détaillés dans le chapitre IV, ont été abordés.

Tout d'abord, les systèmes réalisés constituent un pas en avant dans la conception de SMA pour le contrôle, en conférant à ce domaine relativement jeune une approche possédant des caractéristiques d'adaptation très peu répandues actuellement. Que ce soit Malachite et sa définition du contrôle à l'aide d'un ensemble de comportements complétant un mécanisme de modélisation existant, ou Obsidian et son apprentissage du contrôle selon la définition de contextes, ces deux systèmes bénéficient d'une généricité importante faisant défaut aux systèmes issus de la littérature. Cette généricité confère également à ce travail un aspect réutilisable non négligeable : grâce à la diversité des problématiques auxquelles il peut théoriquement être appliqué, la majorité des notions et des architectures des agents présentés peut fournir la base de travaux futurs.

Concernant le contrôle en général, et son application biologique en particulier, cette thèse présente une approche novatrice dans la place conférée à la modélisation du système à contrôler. La création de ce modèle n'est plus une étape incontournable et l'apprentissage d'une politique de contrôle peut être mené à bien sans celui-ci.

Obsidian démontre en effet qu'il est possible de contrôler un système en se focalisant sur la modélisation directe du contrôle. Il est donc superflu d'être en mesure de prédire le comportement du système pour l'ensemble des situations possibles alors que l'on peut juste apprendre comment faire pour le guider vers l'objectif défini. En ce sens, le processus de contextualisation opéré par Obsidian peut être considéré comme un apprentissage spécialisé dirigé par un objectif.

En outre, le système de contrôle utilisé n'est plus une boîte noire telles que peuvent l'être de nombreuses approches issues de l'intelligence artificielle, mais offre une présentation contextuelle du contrôle facilement interprétable. Le système est en mesure d'expliquer pourquoi et dans quel but il effectue ses choix et, par extension, l'utilisateur a accès à ce raisonnement.

Un tel mécanisme de contrôle peut donc éventuellement apprendre en parallèle au déroulement d'un procédé classique supervisé par un opérateur humain, et lui suggérer certaines actions tout en fournissant les détails sur le raisonnement qui a conduit à cette suggestion.

Enfin, l'apport informatique de ces travaux touche également la théorie des AMAS. Cette

thèse apporte un ensemble d'éléments permettant de mieux comprendre comment modéliser la notion d'interdépendance entre certains agents lorsque celle-ci n'est pas identifiable directement. En particulier, Obsidian présente une synchronisation émergente des actions selon les contextes utilisés. Chacun des agents entreprenant une action n'a aucune visibilité des autres actions effectuées au même moment par les autres agents, mais la structure des contextes permet de maintenir la cohérence de leur action simultanée afin d'atteindre un objectif donné.

En cela, la structure de cet AMAS fournit donc des éléments permettant de tenir compte de ces interdépendances indéfinies *a priori*, tout en utilisant leurs caractéristiques dans le processus d'auto-organisation du système au fur et à mesure de leur apprentissage.

VII.2 L'impact de la pluridisciplinarité

L'aspect essentiel de cette thèse réside dans sa pluridisciplinarité. Définie à la base pour traiter une problématique purement biologique, à savoir la modélisation informatique de colonies cellulaires, la réflexion scientifique a fortement évoluée au fil du temps et de la meilleure compréhension que nous possédions (du point de vue informatique) de l'environnement d'application biologique. Cette évolution s'est poursuivie jusqu'à conduire au problème du contrôle de bioprocédés, délaissant donc l'aspect modélisation.

Cela peut sembler paradoxal, mais affiner le sujet afin de répondre à une attente précise, qui est le contrôle de bioprocédés dans des conditions les plus réelles possibles, s'est accompagné d'une ouverture de la problématique au contrôle en général, incluant de ce fait tout un nouveau domaine d'étude. Ainsi, définir un fil conducteur dans la découverte de ces univers bien différents et arriver à abandonner le regard de l'informaticien sur ces diverses approches s'est avéré une tâche particulièrement complexe bien que gratifiante.

L'une des grandes difficultés rencontrées au cours de ce travail réside dans l'acquisition des compétences nécessaires à appréhender les problématiques à la fois du point de vue biologique et du point de vue informatique.

Il s'est ainsi avéré nécessaire d'acquérir un « double langage » informatique et biologique afin de créer un lien entre les différentes informations à manipuler. La signification d'un même terme selon la spécialité peut ainsi varier drastiquement et certaines données initiales, supposées triviales selon le point de vue, peuvent nécessiter une étude détaillée.

Acquérir ce « double langage » ne concerne pas seulement le vocabulaire employé, mais

concerne également la compréhension des outils utilisés dans les différents domaines.

Par exemple, la nature dynamique des données biologiques liée à la diversité des échelles de valeurs utilisées, le traitement des informations et l'utilisation de modèles développés sous Matlab se sont accompagnés de nombreuses contraintes techniques imprévues, en particulier dans leur cohabitation avec le système de contrôle développé en Java.

En ce sens, il est important de s'attarder sur l'application biologique pratique des travaux décrits ici. Leur application à un modèle complexe d'un bioprocédé composé des équations définissant à la fois la physique du bioréacteur ainsi que l'évolution des micro-organismes est bien avancée, mais a malheureusement souffert d'un ensembles de problèmes, tant conceptuels que techniques, ayant retardé leur implémentation.

Le premier problème rencontré l'a été lorsqu'il s'est agit d'appliquer Malachite à ce problème de contrôle d'un bioprocédé réel. Malachite doit se baser sur un modèle du système à contrôler et l'utilisation du modèle du bioprocédé, à la fois en tant que modèle et système à contrôler, semblait pertinente. Les travaux d'extraction des pseudo-modèles reposant sur la décomposition des différentes équations différentielles composant le modèle ont été effectués, mais l'application pratique du contrôle fut stoppée en raison d'une mauvaise compréhension de la réalité biologique se cachant derrière le problème à résoudre. En effet, et il s'agit d'un exemple de la difficulté de manier le « double langage » informatique/biologie, contrôler un tel bioprocédé de cette manière n'a pas de réelle pertinence biologique. Le modèle sur lequel le système devait être conçu n'est pas accessible en pratique lors du contrôle d'un bioprocédé réel et, de ce fait, la signification même de l'expérience, l'application du contrôle dans un exemple du monde réel, devenait invalide.

N'ayant par conséquent aucun modèle sur lequel baser notre système de contrôle et le processus de création automatique de modèle semblant trop complexe à mettre en œuvre dans le temps imparti, une révision des hypothèses de la problématique a été opérée : le contrôle devait être effectué à partir des observations des variables seules, sans recours au modèle du système à contrôler. Cette hypothèse conduisit au développement d'Obsidian et le guida afin de maintenir la pertinence de son application en tant que système de contrôle d'un bioprocédé réel.

Cet exemple souligne donc cette difficulté, inhérente à la distance séparant le domaine de la biologie et celui de l'informatique, de saisir d'emblée les tenants et aboutissants d'une situation au carrefour de plusieurs champs de recherche.

En contrepartie, et d'un point de vue purement personnel, cette diversité des connaissances

requises et la recherche des similitudes entre deux mondes aussi différents représente une source d'enrichissement certaine de par la nécessité de jongler entre différentes focalisations permettant d'appréhender de nombreux aspects de ce travail sous un jour différent.

Finalement, une fois l'équilibre trouvé entre informatique et biologie, cette pluridisciplinarité se caractérise par la cohérence de la mise en commun de ces différents domaines. Bien que ceux-ci semblent de prime-abord relativement éloignés, leurs études soulignent une convergence de certaines problématiques liées au contrôle. Et sur ces points particuliers, la fusion des différents points de vue, des différentes approches du problème, peut conduire à l'émergence de solutions innovantes comme en témoigne la conception d'Obsidian.

VII.3 Les perspectives

Cette thèse offre une base solide dans le développement de SMA adaptatifs pour le contrôle. Celle-ci peut cependant bénéficier de certains ajouts, constituant les différentes perspectives d'évolution du travail développé dans ce manuscrit.

Tout d'abord, et comme le souligne la conclusion du chapitre VI, une amélioration de la définition de la fusion des agents contextes me semble une priorité. Celle-ci permettrait en effet une augmentation des capacités d'apprentissage d'Obsidian et, de ce fait, rendrait plus facilement possible son application à des problèmes réels grâce à la possibilité d'apprendre à contrôler un système à l'aide de l'observation de plusieurs exécutions successives. Les données des contextes ne serviraient donc pas seulement les capacités d'adaptation du système de contrôle, mais également sa réutilisabilité ainsi que son optimisation.

Cette étape particulière d'amélioration des mécanismes de fusion rejoint la problématique de persistance de l'apprentissage et de sa remise en question. Elle pourrait ainsi bénéficier d'une étude particulière de ce domaine de l'intelligence artificielle duquel elle rejoint la problématique générale.

La perspective suivante représente la suite logique du travail effectué durant cette thèse, à savoir son application au contrôle d'un système modélisant le fonctionnement d'un bioprocédé réel. Celle-ci permettrait de confronter Obsidian aux derniers points restés en suspens tels que l'impact de la gestion du temps dans le contrôle, par exemple avec la présence de divers délais. Cette application permettra ainsi de valider le passage à l'échelle d'Obsidian, bien que la force de sa conception, qui réside dans le peu d'agents nécessaires à la description d'un problème particulier, plaide en la faveur de ce passage à l'échelle.

Actuellement en cours de réalisation, cette application permettra également de valider la pertinence biologique de l'approche présentée dans cette thèse, que ce soit au niveau des hypothèses ayant guidé le développement d'Obsidian aussi bien qu'au niveau des résultats obtenus soulignant l'intérêt de l'utilisation pratique de tels SMA pour le contrôle de bioprocédés.

Enfin, le dernier point reprend les apports de cette thèse vis-à-vis de la théorie des AMAS et de son application au contrôle à l'aide de la notion de contexte. Cette dernière notion offrant la possibilité de créer un système de contrôle réellement générique, l'appliquer à des problèmes radicalement différents permettrait de l'affiner et d'observer son efficacité dans de nouveaux cadres au sein desquels elle pourrait être confrontée à différentes approches de contrôle, centrées ou non sur l'utilisation de SMA mais fournissant une base de comparaison de notre système.

En effet, la définition d'un référentiel commun permettant cette comparaison sur le contrôle de bioprocédés reste pour le moment impossible à cause de l'absence de benchmark permettant de se positionner vis-à-vis des approches usuelles. De plus, les apports spécifiques à l'approche proposée, tels que la réduction du coût et du temps de développement des modèles nécessaires au contrôle, sont des données difficilement quantifiables et fortement dépendantes des problèmes à résoudre, et nécessitent donc un retour de la part d'un utilisateur final expert du domaine afin d'être précisément évalués.

L'amélioration de la solution proposée dans cette thèse passe donc par deux points distincts. Tout d'abord, il est nécessaire de comparer la solution proposée à des systèmes existants, sur un large panel de problèmes différents, afin d'évaluer son efficacité et de confirmer sa généricité. Enfin, il s'agit d'appliquer cette solution au contrôle de systèmes réels, ce qui représente l'étape ultime permettant de valider l'adéquation de son fonctionnement.

Références bibliographiques

[Aastrom 1995] K. J. Aastrom et T. Hagglund. PID Controllers : Theory, Design, and Tuning. Instrument Society of America, Research Triangle Park, NC, 2 édition, 1995. 27

[Astrom 1994] Karl Johan Astrom et Bjorn Wittenmark. Adaptive control. Addison-Wesley Longman Publishing, Boston, MA, USA, 1994. 33

[Baez 2005] J. Baez, T. Stratulat et J. Ferber. *Un modele institutionnel pour sma organisationnel*. Proceedings of Journées Francophones sur les Systemes Multi-Agents (JFSMA 05), Calais-France, 2005. 21

[Bernon 2003] Carole Bernon, Marie-Pierre Gleizes, Sylvain Peyruqueou et Gauthier Picard. *ADELFE, a Methodology for Adaptive Multi-Agent Systems Engineering*. In P Petta, R Tolksdorf, F Zambonelli et S Ossowski, editeurs, International Workshop on Engineering Societies in the Agents World (ESAW), Madrid, Spain, 16/09/2003-17/09/2003, numéro 2577 de LNAI, pages 156–169, http ://www.springerlink.com/, septembre 2003. Springer-Verlag. 79

[Bernon 2005] Carole Bernon, Valerie Camps, Marie-Pierre Gleizes et Gautier Picard. *Engineering Adaptive Multi-agent Systems : the ADELFE Methodology*. In B. Henderson-Sellers et P. Giorgini, editeurs, Agent-Oriented Methodologies, pages 172–202. Idea Group Pub, June 2005. 63

[Bernon 2009] Carole Bernon, Davy Capera, Jean-Pierre Mano, Sylvain Videau et Christine Régis. *Towards Self-Modelling of Metabolic Pathways*. Journal of Biological Physics and Chemistry, vol. 9, no. 1, pages 43–50, mars 2009. 62

[Black 2007] G. Black et V. Vyatkin. *On practical implementation of holonic control principles in baggage handling systems using IEC 61499*. Holonic and Multi-Agent Systems for Manufacturing, pages 314–325, 2007. 22

[Camps 1998] Valérie Camps. *Vers une théorie de l'auto-organisation dans les systèmes multi-agents basée sur la coopération : application à la recherche d'information dans un système d'information répartie.* Thèse de doctorat, Université Paul Sabatier, Toulouse, France, janvier 1998. 60

[Carpenter 2002] J. Carpenter, P. Clifford et P. Fearnhead. *Improved particle filter for nonlinear problems.* In Radar Sonar and Navigation, IEE Proceedings, volume 146, pages 2–7. IET, 2002. 40

[Cockburn 1992] D. Cockburn, L.Z. Varga et NR Jennings. *Cooperating intelligent systems for electricity distribution.* In Proc. Expert Systems 1992. Citeseer, 1992. 44, 46, 48

[Colosimo 2006] B.M. Colosimo et E. Del Castillo. Bayesian process monitoring, control and optimization. Chapman & Hall/CRC, 2006. 39

[Davidsson 2000] P. Davidsson et M. Boman. *Saving energy and providing value added services in intelligent buildings : a MAS approach.* Agent Systems, Mobile Agents, and Applications, pages 79–143, 2000. 45, 46

[Davidsson 2002a] P. Davidsson et F. Wernstedt. *A multi-agent system architecture for coordination of just-in-time production and distribution.* The Knowledge Engineering Review, vol. 17, no. 04, pages 317–329, 2002. 44

[Davidsson 2002b] Paul Davidsson et Fredrik Wernstedt. *Software Agents for Bioprocess Monitoring and Control.* Journal of Chemical Technology and Biotechnology, vol. 77, pages 761–766, 2002. 45

[Demazeau 1995] Y. Demazeau. *From interactions to collective behaviour in agent-based systems.* In In : Proceedings of the 1st. European Conference on Cognitive Science. Saint-Malo. Citeseer, 1995. 20

[Di Caro 2004] G. Di Caro et M. Dorigo. *Ant Colony Optimization and its application to adaptive routing in telecommunication networks.* Unpublished doctoral dissertation, 2004. 24

[Dorigo 2006] M. Dorigo, M. Birattari et T. Stutzle. *Ant colony optimization.* IEEE Computational Intelligence Magazine, vol. 1, no. 4, pages 28–39, 2006. 23

[Erceau 1991] J. Erceau et J. Ferber. *L'intelligence artificielle distribuée.* Recherche, no. 233, pages 750–758, 1991. 20

[Evensen 2003] G. Evensen. *The ensemble Kalman filter : Theoretical formulation and practical implementation.* Ocean dynamics, vol. 53, no. 4, pages 343–367, 2003. 40

[Feldbaum 1961] A.A Feldbaum. *Dual Control Theory I-IV.* Automation Remote Control, vol. 21,22, 1960-1961. 32, 33

[Ferber 1995] J Ferber. Les systèmes multi-agents : vers une intelligence collective. Inter-Editions, 1995. 15, 17

[Ferber 2003] J. Ferber, O. Gutknecht et F. Michel. *From agents to organizations : an organizational view of multi-agent systems.* Agent-Oriented Software Engineering IV, pages 443–459, 2003. 21

[Ferber 2005] J. Ferber, F. Michel et J. Baez. *AGRE : Integrating environments with organizations.* Environments for Multi-agent Systems, pages 48–56, 2005. 21

[Gao 2010] Ying Gao, Katie Kipling, Jarka Glassey, Mark Willis, Gary Montague, Yuhong Zhou et Nigel Titchener-Hooker. *Application of Agent-Based System for Bioprocess Description and Process Improvement* . Biotechnology Progress, vol. 26, no. 3, pages 706–716, May/June 2010. 45, 49, 51, 169

[Gatti 2007] M. Gatti, JE de Vasconcellos et C.J.P. Lucena. *An agent oriented software engineering approach for the adult stem-cell modeling, simulation and visualization.* In Workshop SEAS, João Pessoa, volume 2, 2007. 53

[Georgé 2004] J.P. Georgé. *Résolution de problèmes par émergence-étude d\'un Environnement de Programmation émergente.* PhD thesis, 2004. 58

[Glize 2001] Pierre Glize. *L'Adaptation des Systemes a Fonctionnalite Emergente par Auto-Organisation Cooperative.* PhD thesis, Toulouse, France, juin 2001. 60

[Goldstein 1999] J. Goldstein. *Emergence as a construct : History and issues.* Emergence, vol. 1, no. 1, pages 49–72, 1999. 59

[Guo 2006] Y.N. Guo, J. Cheng, D. Gong et J. Zhang. *A Novel Multi-agent Based Complex Process Control System and Its Application.* Intelligent Control and Automation, pages 319–330, 2006. 44, 48

[Hassas 2003] S. Hassas. *Systèmes complexes à base de multi-agents situés.* University Claude Bernard Lyon, 2003. 23

[Hornik 1993] K. Hornik. *Some new results on neural network approximation.* Neural Networks, vol. 6, no. 8, pages 1069–1072, 1993. 38

[H.P. Whitaker 1958] J. Yamron H.P. Whitaker et A. Kezer. *Design of model reference adaptive control systems for aircraft.* Rapport technique R-164, MIT, 1958. 32

[Hsieh 2002] F.S. Hsieh. *Modeling and control of holonic manufacturing systems based on extended contract net protocol.* In American Control Conference, 2002. Proceedings of the 2002, volume 6, pages 5037–5042. IEEE, 2002. 48

[Huang 2002] Haitao Huang et James B. Riggs. *Comparison of PI and MPC for control of a gas recovery unit.* Journal of Process Control, vol. 12, no. 1, pages 163 – 173, 2002. 34

[Jelasity 2006] M. Jelasity et O. Babaoglu. *T-Man : Gossip-based overlay topology management*. Engineering Self-Organising Systems, pages 1–15, 2006. 24

[Jennings 1991] N.R. Jennings et Q. Mary. ARCHON : An Architecture for Cooperating Systems. Queen Mary and Westfield College, 1991. 46, 47

[Julier 2005] S.J. Julier et J.K. Uhlmann. *Unscented filtering and nonlinear estimation*. Proceedings of the IEEE, vol. 92, no. 3, pages 401–422, 2005. 40

[Kennedy 2006] J. Kennedy. *Swarm intelligence*. Handbook of Nature-Inspired and Innovative Computing, pages 187–219, 2006. 23

[Koestler 1979] Arthur Koestler. The ghost in the machine. Hutchinson, London, the danube ed., 2. impr édition, 1979. 22

[Konstantinov 1993] Konstantin B. Konstantinov, Robert Aarts et Toshiomi Yoshida. Bioprocess design and control, volume 48/1993, chapitre Expert systems in bioprocess control : Requisite features, pages 169–191. Springer, 1993. 41

[Langton 1990] C.G. Langton. *Computation at the edge of chaos : Phase transitions and emergent computation*. Physica D : Nonlinear Phenomena, vol. 42, no. 1-3, pages 12–37, 1990. 59

[Lee 1990] C.C. Lee. *Fuzzy logic in control systems : fuzzy logic controller. I*. IEEE Transactions on systems, man and cybernetics, vol. 20, no. 2, pages 404–418, 1990. 37

[Lee 1994] J.H. Lee et N.L. Ricker. *Extended Kalman filter based nonlinear model predictive control*. Industrial & Engineering Chemistry Research, vol. 33, no. 6, pages 1530–1541, 1994. 39

[Lee 1998] Y. Lee, S. Park et M. Lee. *PID controller tuning to obtain desired closed loop responses for cascade control systems*. Ind. Eng. Chem. Res, vol. 37, no. 5, pages 1859–1865, 1998. 31

[Leriche 2006] Sébastien Leriche. *Architectures à composants et agents pour la conception d'applications réparties adaptables*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, décembre 2006. 64

[Lotka 1910] Alfred J. Lotka. *Contribution to the Theory of Periodic Reactions*. The Journal of Physical Chemistry, vol. 14, no. 3, pages 271–274, 1910. 119

[Luck 2005] M. Luck, P. McBurney, O. Shehory et S. Willmott. *Agent technology : computing as interaction (a roadmap for agent based computing)*. 2005. 17

[Miller 1995] W.T. Miller, R.S. Sutton et P.J. Werbos. Neural networks for control. MIT press, 1995. 37

[Nikolaou 2001] M. Nikolaou. *Model predictive controllers : a critical synthesis of theory and industrial needs*. Advances in Chemical Engineering, vol. 26, pages 131–204, 2001. 34

[Nwana 1996] Hyacinth S. Nwana et Martlesham Heath. *Software Agents : An Overview*, 1996. 19, 169

[Parunak 2000] H.V.D. Parunak. *Industrial and Practical Applications of DAI*. Multiagent systems : a modern approach to distributed artificial intelligence, page 377, 2000. 25, 45, 54

[Picard 2004] Gauthier Picard. *Méthodologie de développement de systèmes multi-agents adaptatifs et conception de logiciels à fonctionnalité émergente*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, décembre 2004. 63

[Rosenblatt 1958] F. Rosenblatt. *The perceptron : A probabilistic model for information storage and organization in the brain*. Psychological review, vol. 65, no. 6, pages 386–408, 1958. 38

[Rougemaille 2008] Sylvain Rougemaille. *Ingénierie des systèmes multi-agents adaptatifs dirigée par les modèles*. Thèse de doctorat, Université Paul Sabatier, Toulouse, France, octobre 2008. 64

[Russell 2009] S.J. Russell et P. Norvig. Artificial intelligence : a modern approach. Prentice hall, 2009. 64, 76

[Serugendo 2005] G.D.M. Serugendo, M.P. Gleizes et A. Karageorgos. *Self-organisation and Emergence in Multi-Agent Systems*. The Knowledge Engineering Review, vol. 20, no. 2, pages 165–189, 2005. 23

[Shmygelska 2005] A. Shmygelska et H.H. Hoos. *An ant colony optimisation algorithm for the 2 D and 3 D hydrophobic polar protein folding problem*. BMC bioinformatics, vol. 6, no. 1, page 30, 2005. 24

[Soderstrom 1988] T. Soderstrom et P. Stoica. *System Identification*, 1988. 32

[Stengel 2002] R.F. Stengel. *Intelligent failure-tolerant control*. Control Systems Magazine, IEEE, vol. 11, no. 4, pages 14–23, 2002. 37

[Tyreus 1992] B.D. Tyreus et W.L. Luyben. *Tuning PI controllers for integrator/dead time processes*. Industrial & Engineering Chemistry Research, vol. 31, no. 11, pages 2625–2628, 1992. 29

[Van de Vijver 1997] G. Van de Vijver. *Emergence et explication*. Intellectica, vol. 25, no. 7-23, page 22, 1997. 59

[Versteegh 2010] F. Versteegh, M.A. Salido et A. Giret. *A holonic architecture for the global road transportation system*. Journal of Intelligent Manufacturing, vol. 21, no. 1, pages 133–144, 2010. 22

[Visioli 2001] A. Visioli. *Tuning of PID controllers with fuzzy logic*. IEE Proceedings - Control Theory and Applications, vol. 148, no. 1, pages 1–8, 2001. 31, 43

[Webb 2006] K. Webb et T. White. *Cell modeling with reusable agent-based formalisms*. Applied Intelligence, vol. 24, no. 2, pages 169–181, 2006. 53

[Williams 1989] R.J. Williams et D. Zipser. *A learning algorithm for continually running fully recurrent neural networks*. Neural computation, vol. 1, no. 2, pages 270–280, 1989. 38

[Wittenmark 2002] B. Wittenmark. *Adaptive dual control*. Control Systems, Robotics and Automation, Encyclopedia of Life Support Systems (EOLSS), Developed under the auspices of the UNESCO, 2002. 34

[Wooldridge 1995] Michael J. Wooldridge et Nicholas R. Jennings. *Agent Theories, Architectures, and Languages : A Survey*. vol. 890, pages 1–22, 1995. 16, 17

[Zadeh 2002] L.A. Zadeh. *Fuzzy logic*. Computer, vol. 21, no. 4, pages 83–93, 2002. 42

[Ziegler 1942] J. G. Ziegler et N. B. Nichols. *Optimum Settings for Automatic Controllers*. Transaction of the ASME, vol. 64, pages 759–768, 1942. 29

Annexe

1 Les Adaptive Value Trackers (AVT)

Les AVT ont été conçus, au sein de l'équipe SMAC [1], afin de permettre l'ajustement d'une valeur définie dans un espace à n dimensions. Une entité, telle un agent, qui désire ajuster une valeur donnée utilise un AVT pour lequel il va constituer son environnement. L'utilisation d'AVT est rendue possible grâce à une bibliothèque Java.

Obsidian utilise de nombreux AVT à une dimension afin d'ajuster les paramètres de ses agents. Cette annexe détaille donc plus particulièrement le fonctionnement spécifique de ces AVT à une dimension.

Un AVT a pour objectif de rechercher une valeur réelle v sous les hypothèses suivantes :

- $v \in [v_{min}; v_{max}]$ où v_{min} désigne la borne minimale de l'espace de recherche et v_{max} sa borne maximale.
- La valeur v est dynamique et, de ce fait, susceptible d'évoluer à tout moment.
- La seule information permettant de trouver la valeur v est un « feedback » de l'environnement appartenant à l'un des trois types suivants : « plus grand », « plus petit » ou « convenable ».

À partir de cet ensemble d'hypothèses, un AVT suit le processus de résolution itératif décrit ci-après.

Soit $v_t \in [v_{min}; v_{max}]$ la valeur proposée pour v par l'AVT à l'instant t.

- L'AVT qui recherche la valeur envoie à son environnement (et donc à l'entité désirant ajuster sa valeur) sa valeur courante v_t.

1. Sylvain Lemouzy, Systèmes interactifs auto-adaptatifs par systèmes multi-agents auto-organisateurs : application à la personnalisation de l'accès à l'information, Thèse de l'université de Toulouse, 2011 (à paraître)

– En fonction des résultats obtenus avec cette valeur, l'environnement retourne un feedback à l'AVT.
– En fonction de ce feedback, l'AVT modifie sa valeur d'un pas Δ : $v_{t+1} = v_t + \Delta$.

Le problème peut alors être reformulé comme la recherche du pas Δ permettant d'ajuster au mieux la valeur v_t par rapport à la valeur recherchée.

Afin de le résoudre, un algorithme se basant sur les éléments suivants a été implémenté :
– Il suffit de connaître la valeur Δ_{t-1} pour prendre en compte les dernières contraintes.
– Il suffit de connaître le tout dernier feedback afin de déterminer s'il y a lieu « d'accélérer » ou de « ralentir » l'ajustement de la valeur de Δ.

Cet algorithme permet de détailler les valeurs que doivent prendre Δ_{t+1} et v_{t+1} à partir de Δ_t , v_t et de l'évolution des feedbacks reçus selon la figure A.1 pour laquelle les variables sont :

– $v_t \in [v_{\min}; v_{\max}]$, la valeur choisie par l'AVT à l'instant t ;
– $\Delta_t \in [\Delta_{\min}; \Delta_{\max}]$, la valeur du pas d'évolution de v_t ;
– $Fb_t \in \{\uparrow; \downarrow; \sim\}$, la valeur du feedback reçu à l'instant t (où $\uparrow$ désigne le feedback « plus grand » ; $\downarrow$, le feedback « plus petit » et $\sim$, le feedback « convenable ») ;
– λ_a, le coefficient d'accélération ($\lambda_a > 1$) ;
– λ_d, le coefficient de décélération ($0 < \lambda_d < 1$).

		Fb_t		
		$\uparrow$	$\downarrow$	$\sim$
Fb_{t-1}	$\uparrow$	$\Delta_t = \Delta_{t-1} \cdot \lambda_a$ $v_{t+1} = v_t + \Delta_t$	$\Delta_t = \Delta_{t-1} \cdot \lambda_d$ $v_{t+1} = v_t - \Delta_t$	$\Delta_t = \Delta_{t-1} \cdot \lambda_d$ $v_{t+1} = v_t$
	$\downarrow$	$\Delta_t = \Delta_{t-1} \cdot \lambda_d$ $v_{t+1} = v_t + \Delta_t$	$\Delta_t = \Delta_{t-1} \cdot \lambda_a$ $v_{t+1} = v_t - \Delta_t$	$\Delta_t = \Delta_{t-1} \cdot \lambda_d$ $v_{t+1} = v_t$
	$\sim$	$\Delta_t = \Delta_{t-1}$ $v_{t+1} = v_t + \Delta_t$	$\Delta_t = \Delta_{t-1}$ $v_{t+1} = v_t - \Delta_t$	$\Delta_t = \Delta_{t-1} \cdot \lambda_d$ $v_{t+1} = v_t$

Figure A.1 – Comportement d'ajustement d'un AVT

Si nous ne possédons aucune information permettant de déterminer *a priori* les valeurs de λ_a et λ_d, celles-ci sont alors initialisées empiriquement de la manière suivante : $\lambda_a = 2$; $\lambda_d = \frac{1}{3}$

Considérons l'exemple suivant : un AVT est chargé de l'ajustement d'une variable représentant la borne inférieure d'une plage d'entrée d'un agent contexte. Cet AVT a reçu à l'instant

$t-1$ un feedback $\downarrow$, information à partir de laquelle l'AVT calcule une valeur v_t. L'environnement lui transmet alors, à l'instant t, un nouveau feedback, cette fois-ci $\uparrow$. La conséquence de ces deux feedbacks successifs est le calcul de $\Delta_t = \Delta_{t-1}.\lambda_d$ correspondant à un ralentissement de la vitesse de variation Δ_t liée au changement du sens du feedback. Ainsi, le calcul $v_{t+1} = v_t + \Delta_t$ fournissant la valeur choisie par l'AVT pour v_{t+1} tient compte de ce ralentissement.

Si à l'instant $t+1$, un nouveau feedback $\uparrow$ est reçu, il confirmera alors la tendance initiée à l'instant t et calculera une valeur d'ajustement $\Delta_{t+1} = \Delta_t.\lambda_a$ tenant compte du coefficient d'accélération λ_a et choisira comme valeur $v_{t+2} = v_{t+1} + \Delta_t$.

Ce fonctionnement permet d'ajuster les différentes variables composant les agents d'Obsidian.

2 Extrait du modèle du bioprocédé

$function[dy, pH] = cinetique(t, Y)$

global grad ParPIDpH ParAer ParFerm ParTemp ParPIDAer ParAlim ParAF fluxCO2 fluxO2 parametres kLaPhy ;

acquisition = 0 ;

AF=0 ;

Y(find(Y<0))=0.0000000000001 ;

pH=calculpHexo(Y(2),Y(15)+Y(16),Y(32)/60) ; % calcul du pH en fonction de la concentration en ion ammonium

[ry, kF]=ModeleBio(t,Y,pH) ;

if Y(13) <1e-9

ry(4)=0 ;

end

if isempty(fluxCO2)

fluxCO2= -ry(5) ;

end

if isempty(fluxO2)

fluxO2= -ry(4) ;

end

if isempty(parametres)

kLaPhy ;

else

kLaPhy = parametres(end,end) ;

end

```
if t >= ParPIDAer(9)
acquisition = 1 ;
kLaPhy = PIDAer(Y,t,ParPIDAer(9),'Manuel') ; % Calcul de l'action regulation pO2( en fonction de pO2 mesuré)
end
if t >= ParPIDpH(10)
acquisition = 1 ;
ParPIDpH(1) = PIDpH(pH,t,ParPIDpH(10) ) ; % Calcul de l'action régulation pH ( en fonction du pH mesuré)
end
if t >= ParAF(4)
acquisition = 1 ;
AF = PIDAF(Y(28),Y(9),t) ; % Calcul de l'action antimousses
end
if t >= ParAlim(8)
acquisition = 1 ;
ParAlim(10)= PIDAlim(Y(14),'batch',Y(21),ry(5),Y(32),t) ;
end
if t >= ParTemp(6)
acquisition = 1 ;
[ParTemp(29),ParTemp(30)]= PIDTemp(Y(22),t) ;
end
if (acquisition == 1)
```

```
if isempty(parametres)
tref= 0 ;
else
tref = parametres(end,1) ;
end
if t > tref ;
rep=[t ParPIDpH ParAer ParFerm ParTemp ParPIDAer ParAF ParAlim kLaPhy] ;
parametres= [parametres ;rep] ;
end
end
% Caractéristiques transfert
VVM=ParAer(6)/ParFerm(1)/60 ;
VolumeHUp =ParFerm(1)*(ParAer(5)^3.5/(750^3.5+ParAer(5)^3.5)*0.6+0.3) ...
*VVM/(VVM+0.7)/(1+ParFerm(3))*(Y(22)/273.15+1) ...
*(1-0.5*(Y(29)^3/(0.003^3+Y(29)^3))) ; % Volume molaire de rétention gazeuse HoldUp
CstHO2=ParAer(7)*exp(ParAer(8)*(-1/298.15+1/(Y(22)+273.15)))*(1+ParFerm(3)) ; % Constante de Henry O2 temp et pression fermenteur
CstHCO2=ParAer(9)*exp(ParAer(10)*(-1/298.15+1/(Y(22)+273.15)))*(1+ParFerm(3)) ; % Constante de Henry CO2 temp et pression fermenteur
o2th =CstHO2*ParAer(2)+ry(4)/kLaPhy ; % pO2 correspondant au transfert physique
if o2th <0
```

```
o2th=0;
end
kLaO2= Y(8) * (1+ 3*(-ry(4))^(1.5)/((1)^(0.7)+(-ry(4))^(1.5))*(1-o2th/(CstHO2*ParAer(2)))) ...
*(1-0.5*(Y(29)^3/(0.003^3+Y(29)^3))); % accélération du transfert O2 par la réaction / action antimousses
% Action anti-mousses
VolumeMort = (ParFerm(2)-ParFerm(1))/(1+ParFerm(3))*(ParFerm(3)/273.15+1)-Y(9);
TempsMelange = 350/(1+ ParAer(5)+150*VVM)/3600; % A modifier en fonction du nombre de puissance ( heure)
pKCarbonate = exp( ParAer(11)/Y(22)+ParAer(12));
dy(1,1)=ry(1); % Accumulation Biomasse
dy(2,1)= ry(2) + ParPIDpH(1)*ParPIDpH(12)*ParPIDpH(13); % Accumulation NH3
dy(3,1)= ry(3)+Y(21)*ParAlim(1); % Accumulation Substrat
dy(4,1)=-ry(4); % Intg O2 consommé
dy(5,1)=ry(5); % Intg CO2 produit
% Correction pH
dy(6,1)= ParPIDpH(1)*ParPIDpH(12)*ParPIDpH(13); % Mole correcteur pH
dy(7,1)= (pH-Y(7))/(ParPIDpH(11)+ TempsMelange); % pH mesuré
% Traitement des Gaz
dy(8,1)= (kLaPhy-Y(8))/(ParPIDAer(10)+TempsMelange); % variation du kla
dy(9,1)= (VolumeHUp-Y(9))/(ParPIDAer(10)+TempsMelange); % variation du Volume Hold-Up
dy(10)= (kLaO2-Y(10))/(Y(9)/ParAer(6)); % Coefficient global de transfert O2
dy(11)= (Y(8)/3-Y(11))/(Y(9)/ParAer(6)); % Coefficient global de transfert CO2
```

O2s= Y(17) ;

r1 = (CstHO2*ParAer(2)-Y(13))/(CstHO2*O2s-Y(13)) ; % Calcul du gradient de transfert O2 (Moyenne logarithmique)

fluxO2 = Y(10)*((CstHO2*ParAer(2)-Y(13))-(CstHO2*O2s-Y(13)))/(log(r1)) ; % Flux transfert O2

CO2s = Y(18) ;

r2 = (CstHCO2*ParAer(4)-Y(16))/(CstHCO2*CO2s-Y(16)) ;

fluxCO2 = Y(11)*((ParAer(4)*CstHCO2-Y(16))-(CO2s*CstHCO2-Y(16)))/(log(r2)) ; % Flux transfert CO2

DebOut =ParAer(6)-22.413995*(fluxO2+fluxCO2) ; % Calcul du débit théorique de sortie en fonction des production gaz Volume molaire 22.413995 (NL)

dy(12,1)= (DebOut-Y(12))/(1/3600) ; % variation du débit de sortie

dy(13,1)=ry(4)+fluxO2 ; % pO2 dissous en Mole/L

dy(14,1)=(Y(13)/ParAer(14)*100-Y(14))/ParAer(13) ; % pO2 mesurée (% de saturation d'étalonnage) cf. pinitialisation

dy(15,1)=ry(5)- (Y(15) - Y(16)* 10 ^(pH-pKCarbonate))/(5/3600) ; % prod carbonate vers equilibre en mole (Mole)

dy(16,1)=(Y(15) - Y(16)* 10 ^(pH-pKCarbonate))/(5/3600)+fluxCO2 ; % pCO2 dissous (Mole)

O2out=(ParAer(6)*ParAer(2)-fluxO2*22.413995)/Y(12) ;

CO2out=(ParAer(6)*ParAer(4)-fluxCO2*22.413995)/Y(12) ;

dy(17,1)=(O2out- Y(17))/(Y(9)/Y(12)) ; % O2 bulle (fraction molaire)

dy(18,1)=(CO2out-Y(18))/(Y(9)/Y(12)) ; % CO2 bulle (fraction molaire)

dy(19,1)= Y(12)/VolumeMort*(Y(17)-Y(19)) ; % O2 mesuré (en fraction molaire)

dy(20,1)= Y(12)/VolumeMort*(Y(18)-Y(20)) ; % CO2 mesuré (en fraction molaire

dy(21,1) =ParAlim(10)*Y(21) ;

% TEMPERATURE

Re = ParTemp(22)*ParTemp(22)*ParAer(5)/60*1000000/.5 ; % Calcul du Nombre de REYNOLDS fct(vitesse agitation)

hm= 0.3*(Re^0.66)*(ParTemp(14)^0.33)*(0.55/ParTemp(15))*4185/3600 ;

% Calcul coef transfert chaleur mout/fermenteur (J/s/m/C)

rQbio = -(ry(1)/27*ParTemp(27)+ry(2)*ParTemp(28)+ry(3)*ParTemp(26)/180+ry(6)*ParTemp(34)/60)*1000/3600 ; % Bilan enthalpique réaction bio (J/s) prod chaleur

dy(22,1) = (rQbio-hm*ParTemp(18)*(Y(22)-Y(23))- ParTemp(24)*ParTemp(18)*(Y(22)-ParTemp(25)))

/(ParTemp(12)*ParFerm(1))*3600 ; % Température mout de fermentation

hp = ParTemp(23) ; % flux de chaleur fermenteur double enveloppe

grad = ((Y(23)-Y(24))+(Y(23)-Y(25)))/2 ; % gradient température

flux = hm*ParTemp(18)*(Y(22)-Y(23)) ... %mout–> paroi

- hp*ParTemp(19)*grad ; % ther –> paroi

dy(23,1) = flux/(ParTemp(13)*ParTemp(21))*3600 ; % Température parois fermenteur

tempsDblEnv = hp*ParTemp(19)*grad/(ParTemp(11)*ParTemp(12))+Y(24) ; % température sortie

grad= Y(24)-Y(26) ;

% gradient echangeur

ech = ParTemp(23)*ParTemp(20)*grad/2 ;% /2 sur coef d'échange thermique

flux = -ech+ ParTemp(11)*ParTemp(12)*(Y(25)-Y(24)) + ParTemp(7)*ParTemp(29) ;

dy(24,1)= flux/(ParTemp(10)*ParTemp(12))*3600 ;

dy(25,1)= (tempsDblEnv-Y(25))/(2/3600) ; % température sortie double enveloppe

% dy(26,1)= (tempsEch-Y(26))/ (2/3600) ; % température sortie echangeur

dy(26,1)= (ParTemp(8)*ParTemp(12)*ParTemp(30)*(-Y(26)+ParTemp(9))+ech)/(0.05*ParTemp(12))*3600 ; % température sortie echangeur

vfoams = Y(27)*VVM/(VVM+.2)*kLaPhy*ParAF(7)*(0.0015^3/(0.0015^3+Y(29)^3)) ; % vitesse formation de mousse

dy(27,1)= kF*Y(1)-vfoams*Y(27)*ParAF(8) ; % concentration agent moussant

dy(28,1)= vfoams-(0.5+ParAF(9)*(Y(29)^3/(0.005^3+Y(29)^3)))*Y(28) ; % volume mousses

dy(29,1)= AF*ParAF(5)*ParAF(6)-Y(1)^(2/3)*(Y(29)-Y(30)) ; % concentration antimousses

dy(30,1) =Y(1)^(2/3)*(Y(29)-Y(30))/10-((ry(1)/Y(1))^2/3+1) *Y(30) ; % concentration antimousses absorbées sur biomasse

dy(31,1) = AF*ParAF(5) ;

dy(32,1) = ry(6) ;

Glossaire

ADELFE La méthode de conception ADELFE (Atelier de DEveloppement de Logiciels à Fonctionnalité Emergente) permet de guider le processus de développement d'un AMAS.

AMAS La théorie AMAS (Adaptive Multi-Agent System), constituant le fondement théorique de cette thèse, détaille la création de SMA centrés sur la notion d'auto-organisation par la coopération des agents. Cette approche permet la conception de systèmes à fonctionnalités émergentes.

AVT Les AVT (Adaptive Value Trackers) sont un outil utilisé afin de permettre l'ajustement d'une valeur définie dans un espace à n dimensions.

Bioprocédé Un bioprocédé est un procédé quelconque utilisant des réactions biologiques afin d'obtenir une production définie. De telles réactions peuvent par exemple utiliser des micro-organismes ou encore des enzymes. Ainsi, un bioprocédé peut aussi bien être une production industrielle de yaourts que le traitement d'eaux usées, en passant par la production chimique pharmaceutique.

Contrôle Désigne dans cette thèse l'ensemble des opérations s'étendant de la surveillance à la commande d'un système.

PID Un régulateur (ou contrôleur) PID (Proportionnel Intégral Dérivé) permet de réguler la valeur d'une variable selon une consigne définie par l'utilisateur. Il s'agit du type de régulateur le plus répandu dans l'industrie.

SNC Une Situation Non Coopérative (ou Non Cooperative Situation) est une situation allant à l'encontre des règles de coopérations que doivent suivre les agents d'un AMAS. Ces situations sont donc les cas problématiques que les agents doivent éviter ou résoudre afin d'être conforme à la théorie des AMAS.

Résumé

Résumé Cette thèse a pour cadre le contrôle auto-adaptatif de procédés biologiques et son objectif est de permettre l'application de ce contrôle à une grande variété de problèmes distincts en évitant la phase usuelle de leur modélisation.

Les bioprocédés sont des systèmes complexes, hautement dynamiques et parvenir à les contrôler en vue d'obtenir une production définie se révèle une tâche difficile. De plus, les incertitudes liées aux mesures et le manque de connaissances des réactions biologiques précises se déroulant à l'intérieur même du bioprocédé, font que les méthodes usuelles mises en œuvre pour contrôler un bioprocédé particulier doivent être largement re-calibrées dès qu'il s'agit d'adapter ce contrôle à un bioprocédé différent. L'apport de cette thèse est de proposer une approche informatique de cette problématique, centrée sur l'utilisation de Systèmes Multi-Agents Adaptatifs (AMAS). Les propriétés auto-organisatrices de tels systèmes, ainsi que leur conception centrée sur les comportements locaux, permettent d'appréhender la complexité des procédés biologiques et de fournir un système apte à leur contrôle sans nécessiter d'informations détaillées sur ceux-ci, tout en étant capable de s'adapter à leurs dynamiques.

Deux systèmes multi-agents adaptatifs distincts, répondant chacun à un ensemble de contraintes différentes, ont permis d'étudier la faisabilité et l'apport de cette approche au domaine du contrôle de procédés. L'aboutissement est un modèle AMAS générique qui est associé à chaque variable contrôlable du procédé afin d'apprendre et de déterminer les actions de contrôle pour atteindre les objectifs définis par l'utilisateur. Les agents observent en temps réel l'évolution des variables du procédé pour en extraire les informations nécessaires à son contrôle. Le système de contrôle établit alors une contextualisation des contrôles à appliquer qui devient indépendante de la connaissance du bioprocédé.

Ce contrôle auto-adaptatif a été évalué sur un ensemble de problèmes de contrôle de systèmes dynamiques, notamment celui d'un bioprocédé simulé, et les résultats obtenus sont analysés dans ce mémoire.
Mots clés : Systèmes multi-agents, adaptation, émergence, contrôle, bioprocédés

Abstract

Abstract This work aims at creating a self-adaptive approach to control bioprocesses while still being generic enough to be applied on a wide range of problems without relying on their modeling.

Bioprocesses are complex and highly dynamic systems, and controlling them to reach a user-defined objective is a difficult task. Furthermore, the limited amount of available measures and the lack of a precise biological knowledge of what is happening inside the bioprocess lead usual control approaches to be recalibrated before being applied on a different bioprocess.

The benefit of this work is a computer-based approach relying on the Adaptive Multi-Agent Systems (AMAS) theory. The objectives of adaptability and genericity are reached thanks to the self-organization of such systems and their design focusing on the local behavior of the agents, dealing with the complexity of such a problem without needing an extensive amount of information on the system to control.

Two distinct multi-agent systems, defined by a specific set of constraints, were created to study the feasibility and benefits of such an approach. The outcome is a generic AMAS model, associated with each controllable variable, which learns the actions to apply in order to fulfill user-defined objectives. Agents observe in real-time the evolution of the variables and extract the information needed to control the target system. The AMAS is then able to contextualize the control without modeling the behavior of the system to control itself.

This self-adaptive control was evaluated on a set of problems involving the control of dynamic systems, such as bioprocesses, and the associated results are analyzed.
Keywords : Multi-agents systems, adaptation, emergence, control, bioprocesses

Abstract

Table des figures

Table des figures

Liste des tableaux