

N° d'ordre :

Thèse

Présentée pour obtenir

LE TITRE DE DOCTEUR
DE L'INSTITUT NATIONAL POLYTECHNIQUE DE TOULOUSE

École doctorale : Systèmes

Spécialité : Systèmes Industriels

Par **Anton Mitsiouk**

Titre de la thèse **Contribution à l'Optimisation des Systèmes Dynamiques :
Application au Génie des Procédés.**

Soutenue le 12 Juillet 2007 devant le jury composé de :

M.	RUIZ Jean Michel	Président, Rapporteur
M.	AVDOSHHN Sergey	Directeur de thèse
M.	LE LANN Jean Marc	Directeur de thèse
M.	RENEAUME Jean Michel	Rapporteur
M.	SERENDINSKI Avraam	Membre
M.	CARILLO LEROUX Galo	Membre
M.	NEGNY Stéphane	Membre

TABLE DES MATIERES

Résumés :.....p 1

Introduction Générale.....p 9

Chapitre 1 :

Modélisation et Simulation des Systèmes Dynamiques Hybrides.....p 17

I Introduction.....p 19

II Du Continu vers les Systèmes Dynamiques Hybrides.....p 20

II.1 Les modèles à Equations Différentielles et Algébriques.....p 20

II.2 Les Bond Graphs.....p 21

II.3 Extension des Modèles Continus.....p 27

III Des Evénements discrets vers les Systèmes Dynamiques Hybrides....p 28

III.1 Les Automates à états finis.....p 29

III.2 Le Grafcet.....p 30

III.3 Les Réseaux de Petri.....p 31

III.4 Les Statecharts.....p 33

III.5 Extension des Modèles à Evénements Discret.....p 34

IV Modélisation des Systèmes Dynamiques Hybrides.....p 36

IV.1 Les Modèles Mixtes.....p 37

IV.2 Simulation des Systèmes Dynamiques Hybrides.....p 39

IV.3 Classification des Outils de Simulation.....p 40

V Conclusion.....p 43

Chapitre 2 :

Méthodes d'Optimisation Dynamique.....p 45

I Introduction.....p 49

I.1 La Plate-Forme PrODHyS.....p 50

I.2 Le Noyau Numérique.....p 53

I.3 Conclusion.....p 56

II Le problème de Contrôle Optimal.....p 57

II.1 Définition.....p 57

II.2 Principe du Maximum de Pontryagin.....p 58

III Les Méthodes indirectes.....p 60

III.1 Méthode de tir simple.....p 60

III.2 Méthode de tir multiple.....	p 61
III.3 Autres Méthodes.....	p 62
III.4 Conclusion.....	p 63
IV Méthodes directes.....	p 65
IV.1 Approche séquentielle.....	p 66
IV.2 Approche Simultanée.....	p 69
V Programmation Dynamique Itérative.....	p 72
VI Conclusion.....	p 75

Chapitre 3 :

Présentation de la Méthode.....	p 77
--	-------------

I Introduction.....	p 81
II Présentation Générale de la méthode.....	p 82
II.1 Discrétisation de l'horizon de temps.....	p 82
II.2 Méthode de Programmation Quadratique Successive.....	p 85
III Résolution Numérique.....	p 87
III.1 Analyse de Sensibilités.....	p 87
III.2 Méthode d'intégration du système global.....	p 92
III.3 Paramétrisation du contrôle.....	p 97
III.4 Les contraintes.....	p 99
IV Conclusion.....	p 102

Chapitre 4 :

Illustration en Génie des Procédés.....	p 105
--	--------------

I Introduction.....	p 107
II Réacteur Fed Batch.....	p 108
II.1 Le modèle.....	p 108
II.2 Résultats.....	p 109
III Optimisation d'un réacteur batch.....	p 110
III.1 Modélisation.....	p 111
III.2 Résultats.....	p 113
III.3 Résolution avec Contrainte de Température.....	p 114
IV Identification Paramétrique.....	p 116
IV.1 Contexte de l'étude.....	p 116
IV.2 Modélisation.....	p 119
IV.3 Simulation et Résolution.....	p 123
IV.4 Résultats.....	p 124
V Conclusion.....	p 126

Conclusion Générale.....p 127

ANNEXES.....p 133

Annexe 1.....p 135

Annexe 2.....p 138

Annexe 3.....p 140

Références Bibliographiques.....p 143

Contribution à l'Optimisation des Systèmes Dynamiques : Application au Génie des Procédés

Résumé :

Ces travaux présentent une méthode d'optimisation dynamique pour le contrôle optimal en génie des procédés. Plus généralement, ils s'inscrivent dans une thématique plus vaste, concernant l'optimisation des systèmes dynamiques hybrides.

La méthode développée repose sur une méthode hybride entre deux méthodes directes : approches séquentielles et simultanées. Cette méthode nommée Méthode directe à tirs multiples, se caractérise par un découpage de l'horizon de temps en sous intervalles d'intégration et par une réduction de l'espace de recherche des fonctions de contrôle par une paramétrisation de ces dernières. Contrairement aux méthodes séquentielles, la méthode présentée permet de prendre en compte facilement des contraintes sur les variables d'état, ce qui représente une classe importante de problème en Génie des Procédés. De plus, la taille du problème d'optimisation reste acceptable car seules les fonctions de contrôle sont paramétrisées. Toutefois, ce problème d'optimisation est plus complexe que celui des approches séquentielles, à cause de la mise en œuvre de contraintes de continuité (sur les variables d'état et de contrôle) entre les différents sous intervalles issus de la subdivisions du temps.

Cette méthode est intégrée dans le noyau numérique de la plate forme PrODHyS. Son utilisation est illustrée au travers de trois exemples en génie des procédés.

Mots Clés : Optimisation Dynamique, Contrôle Optimal, Méthode Directe à tirs multiples, Simulateur dynamique hybride de procédés (PrODHyS), Systèmes EDA.

Summary :

These works present a method for dynamic optimization and more precisely for optimal control in Chemical Engineering. More generally, they are included in a vaster research which deals with optimization of hybrid dynamic systems.

The developed method is a hybrid one between two direct methods: sequential and simultaneous approaches. In this method, called multiple shooting type techniques (direct method), the original time interval is divided into multiple shooting intervals with DAE solved numerically on each subintervals. In order to reduce the research space for control function, we use control parameterization. With this method we can take into account state path constraints, which include an important number of problems in chemical engineering (which is not feasible with sequential approach). Moreover, the optimization problem size is less important than in the simultaneous approach because only the control functions are parameterized. The optimization problem is more complex because of continuity constraints imposed across subintervals (on state and control variables).

This method is integrated within the numerical kernel of the platform PrODHyS. It is illustrated with three chemical engineering examples.

Keywords: Dynamic Optimization, Optimal Control, Multiple shooting type techniques, Hybrid dynamic process simulator, DAE systems.

Научные руководители

к.т.н., профессор Авдошин С.М.,

д.т.н., профессор Леланн Ж.М.,

Аспирант ММЧМКП*

Мицюк А.А.

Специальность: 05.13.18.

Разработка и исследование методов оптимизации динамических гибридных систем

Резюме диссертации на соискание ученой степени кандидата технических наук

До недавнего времени при моделировании, анализе и оптимизации промышленных процессов в химической, пищевой и фармацевтической индустрии мы разделяли их на непрерывные и дискретные. Однако, в реальном мире непрерывные и дискретные процессы существуют вместе, активно взаимодействуя. Например, бассейн с двумя трубами обладает сложным поведением, поскольку его поведение при переполнении качественно отличается от его поведения при нормальном уровне и при опустошении. Такой тип сложного поведения можно реализовать, если описать всю совокупность допустимых, простых, в некотором смысле, частных поведений (возможно, это будет иерархическая структура), и указать правила переключения с одного поведения на другое. Организованная таким образом, сложная динамическая система в каждый конкретный момент времени ведет себя как некоторая простая динамическая система. Например, бассейн может быть представлен как совокупность трех простых динамических систем: "Нормальный уровень", "Переполненный бассейн" и "Пустой бассейн".

Каждое конкретное поведение можно отождествить со значением некоторой дискретной переменной, а мгновенные переключения текущего поведения - с дискретными событиями. Для передачи информации о дискретных событиях в другие блоки используют специальные переменные - сигналы. Набор дискретных состояний вместе с условиями переходов из одного состояния в другое образует обычное дискретное поведение. В моменты переходов могут происходить мгновенные скачкообразные изменения значений переменных. Поскольку в каждом из дискретных состояний элементарный блок ведет себя как некоторая непрерывная система, то поведение блока в целом является непрерывно-дискретным или гибридным. Граф переходов, узлам которого приписаны некоторые непрерывные отображения, а дугам – условия переходов и выполняемые действия, называется гибридным автоматом.

Системы, в которых взаимодействие, дискретно и непрерывно изменяющихся переменных является настолько сложным, что его невозможно изучать по отдельности, разделив систему на две подсистемы, получили название динамических гибридных систем (ДГС).

В последние годы изучение динамических гибридных систем получило новый импульс. Важность развития областей применения ДГС, таких как ТЭК, пищевое и фармацевтическое производство, транспортные системы, коммуникационные протоколы предопределили интерес научного сообщества к этой исследовательской области. Тремя главными направлениями изучения ДГС являются моделирование, анализ и управление ДГС.

Целью представленной работы были разработка и исследование методов оптимизации динамических гибридных систем. Оптимизация, а особенно оптимальное управление ДГС, является очень перспективным научно-

практическим направлением, в котором работаю научно-исследовательские лаборатории во многих странах мира. Представленная работа велась в тесном взаимодействии с Национальным Политехническим Институтом Тулузы, Франция и Тулузской Лабораторией Химических Технологий.

Термин оптимизации ДГС может относиться к двум различным проблемам, каждая из которых соответствует своему уровню оптимизации:

На первом уровне проблема оптимизации непрерывной (или непрерывных) подсистемы. Эту проблему часто называют проблемой оптимального контроля,

Второй тип проблемы относится к более высокому типу оптимизации. Его целью является оптимизация порядка непрерывных последовательностей.

Представленная работа посвящена первому типу проблемы, а именно, проблеме оптимального контроля.

На начальном этапе работа сосредоточилась на моделях и средствах имитационного моделирования, часто используемых в области гибридных динамических систем. Было важно определить характеристики и свойства, позволяющие их классифицировать. К свойствам моделей можно отнести модульности, причинности, однозначное представление непрерывных и дискретных подсистем, способности управлять общей переменной структурой алгебро - дифференциальной системы. Средства имитационного моделирования характеризуются модульным или глобальным характером имитационного моделирования, способностью решать обычные дифференциальные уравнения (ДУ), ДУ в частных производных, алгебраические дифференциальные уравнения, использованием алгоритма понижения индекса, способностью управлять событиями и явно или неявно переинициализировать переменные.

Сферой применения полученных результатов было выбрано управление процессами. В этой области, так называемое "гибридное" моделирование имеет большую значимость, так как оно отвечает потребности моделировать все более и более сложные реальные системы, упрощение динамик которых ведет к разрывам в их представлении. Прекрасными примерами таких систем являются пакетные процессы, переходные фазы операций пуска, остановки, очистки, изменения физических состояний веществ.

Одной из задач обзора ДГС являлось ограничение области исследований по отношению к системам с непрерывной и дискретной динамиками. После обсуждения различных аспектов моделирование были представлены разнообразные формализмы, позволяющие создавать модели ДГС необходимой точности. формализмы отличаются друг от друга техникой моделирования, которая основывается либо на подходе, стремящемся расширить дискретную или непрерывную модели, либо на подходе, который изначально стремится использовать смешанную модель. Исследования показали, что техника моделирования, базирующаяся на расширении формализма, изначально

предусмотренного для дискретной или непрерывной систем серьезно ограничена. С одной стороны, она предпочитает одну из двух динамик системы за счет другой; в то же время у реальных систем сложное поведение, которое требует точного моделирования и непрерывной, и дискретной подсистем. С другой стороны, она приносит мало ясности и читабельности в разграничении между непрерывным и дискретным взглядами на систему. К тому же она не соблюдает некоторых принципов эволюции маркировки, классически приписанных сетям Петри: это относится, например, к непрерывным сетям Петри, в которых два перехода, находящиеся в конфликте, могут быть преодоленными одновременно. Наконец, использование этой техники часто имеет результатом создание моделей со специфическими характеристиками, которые имеют очень ограниченное применение.

Напротив, моделирование, основывающееся на смешанном подходе является более универсальной и обладает большим потенциалом моделирования в каждой из областей (непрерывной и дискретной). К тому же, она позволяет использовать модели высокого уровня для моделирования событий.

Обзор ДГС, предложенный в первой главе, позволяет с большей легкостью взяться за проблематику исследования, которая состоит в оптимизации ДГС. Конечной целью диссертации является добавление модуля оптимизации к платформе PRODHYS, разработанной в нашей лаборатории в Тулузе и представленной в следующей главе.

До описания предлагаемого метода в диссертации описаны часто встречающиеся в области динамической оптимизации методы. Они разделены на две большие категории: косвенные методы и прямые методы. Первые вытекают из применения принципа максимума Понтрягина. Что касается вторых, то они превращают проблему оптимального контроля в проблему нелинейной оптимизации с ограниченным размером. Вторая категория может быть классифицирована еще раз в зависимости от выбранного уровня: частичная дискретизация проблемы (последовательный подход) или общая дискретизация (синхронный подход).

В следующей главе изучены наиболее распространенные подходы к решению проблемы динамической оптимизации. Первые стратегии решения проблем динамической оптимизации использовали косвенные методы, основанные на применении принципа максимума Понтрягина. Для корректного использования этих методов необходимо уделить особое внимание предельным условиям. В большинстве случаев, на переменные состояния накладываются начальные условия, а на вспомогательные - окончательных условий. Эта проблема предельных значений на двух концах может быть решена методом простой стрельбы или методом многократной стрельбы. Между тем, когда надо принять во внимание ограничения-неравенства переменных состояния, решение

становится более сложным: становится особенно трудным найти связанные первоначальные значения для переменных состояния и вспомогательных переменных.

Методы, основанные на решении проблемы нелинейного программирования, могут разделиться на две главные категории: последовательные методы и синхронные методы. Методы, основанные на решении проблемы нелинейного программирования могут разделиться на две главные категории: последовательные методы и синхронные методы. В проблемах первой категории мы дискретизируем только переменные контроля (часто в полиномиальной форме). В этом случае оптимизация осуществляется по отношению к коэффициентам многочленов. Со начальными условиями, связанными со всеми параметрами контроля, интегрирование системы алгебро-дифференциальных уравнений (АДУ) включено в решение проблемы нелинейного программирования. В результате каждой итерации алгоритма оптимизации, пересчитываются коэффициенты многочленов контроля, и система АДУ интегрируется с использованием этих новых значений. Подобные последовательные методы относительно легки в реализации и приносят убедительные результаты (главным образом, если интегратор системы АДУ и метод оптимизации надежны). Напротив, систематическое интегрирование системы АДУ при каждой итерации может стать ресурсоемкой в том, что касается времени подсчета, главным образом в случае систем большого размера. Кроме того, принятие в расчет ограничений-неравенств переменных состояния осуществляется только приблизительно благодаря штрафным функциям.

При использовании синхронных методов дискретизируются одновременно и переменные контроля, и переменные состояния. Следовательно, эти подходы ведут к проблемам линейного программирования большого размера, которые требуют эффективных методов решения. По итогам дискретизации переменных состояния решение системы АДУ прямо связано с проблемой оптимизации. В финале, система АДУ решается лишь раз, в оптимальной точке, что позволяет избежать больших временных затрат на расчеты, а также на промежуточные решения. Эти методы надежны и могут справиться с нестабильными значениями входных переменных, что бывает полезным для некоторых систем. Более того, принятие в расчет ограничений переменных состояния и контроля осуществлено на том же уровне дискретизации, что и дискретизация переменных системы АДУ.

Еще одной важной категорией методов решения проблемы динамической оптимизации являются прямые методы многократной стрельбы. Они могут рассматриваться как промежуточные между двумя типами методов, описанными ранее. Их часто объединяют с синхронными техниками, но по сравнению с ними они обладают многочисленными преимуществами, унаследованными от последовательных методов. В принципе, эти методы разделяют горизонт времени на под-интервалы, на которых интегрируется система АДУ (отдельно на

каждом интервале). В то же время переменные контроля обрабатываются идентично тому, как это происходит в последовательных методах.

Предложенный и реализованный метод относится именно к такому промежуточному подходу. Он разделяет горизонт времени на под-интервалы, в которых система интегрируются дифференциальные и алгебраических уравнения.

Он станет отличным дополнением к платформе PRODHYS. Этот элемент позволит осуществить оптимизацию динамической системы (решить проблему оптимального контроля, какой бы ни была проблема). В области управления процессами, присутствие в многочисленных моделях ограничений-неравенств заставляет нас из многочисленных методов решения этого класса проблем использовать прямые. В этой категории методов можно выделить два главных подхода, которые главным образом отличаются уровнем дискретизации. Синхронный подход позволяет принять в расчет ограничения переменных состояния, однако приходится решать систему очень большого размера (необходимость дискретизации переменных состояния и контроля). Учитывая выбранную область применения результатов диссертации, важно научиться работать с ограничениями переменных состояния (ограничения значений температуры, давления, состава) без масштабного увеличения размера проблемы. Для того, чтобы удовлетворять жестким требованиям к методу решения проблемы, был выбран гибридный между синхронным и последовательным подход. Метод, названный прямым с многократной стрельбой, основан на работах Бок и Плат (1984), возобновленных Сербан и Петцольд (2001), а также Ляйнвебер и др. (2003 а и b). Главная идея этого подхода состоит в том, чтобы для интегрирования разделять горизонт от времени на под-интервалы. За исключением первого под-интервала, начальные условия каждого под-интервала (переменные состояния) включены в множество переменных решения. Между тем, это разбиение требует добавления ограничений на непрерывность переменных состояния. Это объясняется тем, что начальное состояние каждого под-интервала соответствует конечному состоянию предыдущего под-интервала. Хотя система АДУ интегрируется на каждой итерации, в предложенном методе есть возможность получения невыполнимых промежуточных решений, как впрочем, во всех синхронных методах.

Представленный метод был интегрирован в платформу динамического имитационного моделирования PrODHyS для последующего применения для решения реальных проблем. В последней части научной работы представлены три примера использования метода.

Перед решением непосредственно проблемы динамической оптимизации важно осуществить некоторые подготовительные этапы. В первую очередь необходимо провести моделирование проблемы, подготовить модели необходимого уровня сложности и точности, определить целевую функцию и

ограничения переменных состояния и управлений. Ограничения на непрерывность генерируются автоматически. Далее рекомендуется проинтегрировать систему АДУ индивидуально, далее с чувствительностями. Эти предварительные этапы могут оказаться очень информативными и полезными для фазы оптимизации.

В первом рассматривается реактор пакетной подачи. На этом достаточно простом примере продемонстрирован общий метод оптимизации подобных реакторов в случае существования ограничений, которые накладываются на функцию управления. Пример прекрасно иллюстрирует гибкость метода в том, что касается параметризации функций управления, дискретизации горизонта времени и управлений. Еще одним преимуществом метода является возможность автоматически генерировать ограничения на непрерывность и легко включать в модель ограничения переменных управления.

Второй пример посвящен реактору пакетной подачи с заливкой. Он демонстрирует пределы использования метода, связанные в результате учета ограничений переменных состояния. Он обязывает нас либо включать дополнительные переменные и их ограничения в финальной точке, либо, в благоприятном случае, увеличивать число точек дискретизации горизонта времени, что однако ведет к увеличению времени вычислений.

Наконец, третий пример подчеркивает развивающийся характер метода, эффективность его работы в паре моделирование - оптимизация, демонстрирует его возможности по параметрической идентификации в динамическом режиме в составе платформы PRODHYS.

Представленная диссертация открывает многочисленные научные перспективы. Заслуживает безусловного внимания задача принятие в расчет ограничений разного вида, а также оптимизация последовательности смены «непрерывных» режимов в процессе функционирования ДГС.

Introduction Générale

Le génie chimique a connu un essor important grâce à la conception et l'exploitation de procédés de grande capacité de production, bien souvent devenus synonymes de procédés continus produisant en régime permanent. Les complexes pétrochimiques en constituent l'exemple le plus représentatif. La notion de régime permanent a fait les beaux jours du génie chimique, d'ailleurs elle est l'élément à la base de la culture de l'ingénieur de procédés (même de nos jours). Le régime permanent implique qu'aucune grandeur (température, pression...) ne varie dans le temps, lorsque le point de fonctionnement nominal est atteint. Cette vision idéale s'éloigne de la réalité industrielle, où ces grandeurs se stabilisent autour de valeurs moyennes. En effet, de nombreuses perturbations empêchent d'atteindre ce régime permanent : variation de la qualité des matières premières, usure des appareillages...

L'approche du régime permanent se retrouve également comme un axe d'étude important et un domaine d'application privilégié pour la simulation. De nos jours les simulateurs de

procédés en régime permanent sont arrivés à maturité : représentation des opérations unitaires, modélisation des phénomènes physiques et chimiques, robustesse et efficacité des calculs. De plus, les modèles en régime permanent prennent de plus en plus en compte la technologie des appareils.

Industriellement, tout procédé est muni de systèmes de contrôle – régulation permettant de palier les instabilités précitées et de maintenir un état quasi stationnaire. Les méthodes de conception du système contrôle/commande nécessitent de l'information et de la connaissance sur la dynamique du procédé. Il faut déterminer les réponses du procédé en boucle ouverte suite à des perturbations programmées. Ces réponses sont difficiles à quantifier expérimentalement pour des raisons de coût, faisabilité, sécurité... La simulation dynamique de procédés continus permet d'accéder à ce genre d'information. Dans cet exemple, le schéma de contrôle ainsi que les réglages pourront ensuite être expérimentés par des simulations en boucle fermée. Toutefois l'intérêt de la simulation dynamique ne se limite pas exclusivement au contrôle. Elle permet également de simuler le démarrage et l'arrêt d'une installation (phases dangereuses où le pilotage est souvent manuel), former des opérateurs (impact d'une décision : ouverture de vanne etc... sur le procédé), simuler certaines opérations unitaires fonctionnant en régime dynamique (distillation, réaction...), réaliser des études de sécurité (réponses à des pannes, élaboration de scénarios catastrophes...) et bien d'autres cas.

Cependant, l'ouverture du génie chimique vers l'ensemble des procédés de fabrication élargit le spectre d'applications des méthodes classiques. Les procédés de production des industries agroalimentaire, biotechnologique, électronique ou pharmaceutique (liste non exhaustive) peuvent s'analyser avec les concepts d'opérations unitaires, avec les bilans... Dans ces industries, le procédé est rarement exclusivement continu, le mode de production est souvent discontinu avec des productions par lots. La production discontinue est employée pour la fabrication de petits tonnages pour laquelle une mise en œuvre continue demanderait des volumes d'appareillage trop petits, pour des productions délicates demandant un savoir faire dont l'équivalent en continu est complexe, pour des actions périodiques de nettoyage ou des contraintes liées à la sécurité. De plus, un des critères de rentabilité de tels procédés est l'utilisation maximale des appareillages : par souci de réactivité et de diminution des coûts. Ce qui amène des politiques de production très élaborées où il n'est pas rare qu'une unité de production fabrique plusieurs produits.

Ces secteurs d'application génèrent des problèmes originaux dont la modélisation et la simulation sont des étapes indéniables. Dans le cadre de la simulation de ces procédés, les systèmes réels sont généralement caractérisés par une combinaison d'étapes continues (dynamiques ou non) et discrètes. Ces systèmes à double composante comportementale sont appelés Systèmes Hybrides. Plus précisément la dénomination Systèmes Dynamiques Hybrides désigne des systèmes à composante dynamique continue. Cette définition fait

apparaître deux parties dans la modélisation de tels systèmes : le modèle continu et le modèle discret. Le modèle continu traduit la physique de chacune des opérations. Tandis que le modèle discret décrit les trajectoires de la matière (enchaînement d'opérations) au cours de son passage dans l'unité de production et en fonction du temps. Le résultat fourni par ce modèle est une suite de changements d'état exprimée en termes d'activités et d'événements.

La simulation dynamique est un outil précieux comme le démontre son utilisation depuis de nombreuses années dans des domaines comme l'aéronautique, le génie électrique... Bien que son utilisation en génie des procédés soit plus tardive, son intérêt est maintenant bien reconnu. Ces apports peuvent se mesurer au travers de plusieurs critères :

- Augmentation du degré de fiabilité des procédés en réalisant des tests en fonctionnement normal et anormal (sans passer par un pilote)
- Amélioration de la compréhension des régimes transitoires de l'unité (phases de démarrage et d'arrêt)
- Amélioration de la conception et des performances de l'unité (sans recourir à des tests grandeur réelle).
- Informations précieuses sur les bilans matières, énergie et sur l'analyse des sensibilités.

De plus, les ingénieurs se doivent de pouvoir répondre à certaines interrogations découlant des perturbations pouvant intervenir sur le procédé :

- Quels sont les effets des perturbations et de quelles manières se propagent elles ?
- Quel sera le système de contrôle le plus sûr et efficace qui assurera l'action voulue dans les plus brefs délais ?
- Comment effectuer de façon optimale les démarrages et les arrêts ?
- Quelles seront les conséquences d'une panne d'un équipement ?
-

Autant de questions délicates auxquelles la simulation dynamique peut apporter des réponses. Ceci devient crucial car compte tenu des impératifs de productivité et de respect des normes, les unités deviennent de plus en plus complexes et diversifiées.

Outre les applications possibles énoncées précédemment, la simulation dynamique peut également s'appliquer dans le cas de recherche de trajectoires optimales entre deux points de fonctionnement et l'étude, la conception et l'optimisation de stratégies de conduite. Ainsi, l'aspect optimisation des systèmes dynamique revêt une importance particulière.

La simulation dynamique en génie des procédés possède des caractéristiques spécifiques qui engendrent des problèmes difficiles à surmonter :

- Un couplage entre différents types d'équations à traiter : équations algébriques non linéaires et des équations différentielles. Ces systèmes sont dénommés algébro-différentiels. Ce couplage complique la résolution.
- La grande taille des systèmes, il est fréquent d'avoir des milliers d'équations. Toutefois, ces systèmes ont souvent une structure creuse qui est exploitée lors de la résolution.
- Une forte non linéarité. Quasiment tous les systèmes sont non linéaires mais à des degrés divers. La conséquence de cette non linéarité est une aggravation du problème de rigidité.
- La rigidité. Ce phénomène apparaît lorsque les constantes de temps des équations sont très différentes. Des phénomènes ultra rapides côtoient des dynamiques lentes.
- Le traitement des discontinuités. Elles résultent de perturbations extérieures et de phénomènes internes au modèle (changement de phase, activation d'une réaction chimique...)

Ce dernier point démontre qu'il est nécessaire de traiter des événements : soit connus à l'avance, soit liés à des variables d'état du système. Leur détection est cruciale et nécessite des techniques spéciales. Mais ces événements confèrent aux procédés des caractéristiques discrètes qui dans le cadre de la simulation dynamique, sont couplées à des étapes continues pour former des Systèmes Dynamiques Hybrides.

Comme énoncé ci-dessus, la composante discrète peut intervenir lors d'un changement de phases mais ce n'est pas le seul cas. Plus généralement, la modélisation de phénomènes physiques est souvent réalisée pour des plages de valeurs limitées, par conséquent le passage d'une plage à l'autre entraîne une discontinuité (même si la nature présente rarement des discontinuités). De plus, des hypothèses simplificatrices lors de l'écriture de modèle peuvent générer ces discontinuités. De plus en plus souvent, ces discontinuités sont gérées par un formalisme discret permettant ainsi une meilleure lisibilité et d'éviter d'éventuels échecs d'intégration lors du franchissement d'une discontinuité (réinitialisation du système). Toutefois la partie discrète peut se retrouver à un autre niveau. Dans les procédés discontinus (industries pharmaceutique, agroalimentaire...), des recettes de fabrication sont effectuées étapes par étapes. Dans ces ateliers, plusieurs recettes utilisent les mêmes appareils et peuvent être réalisées en parallèle. De plus chaque recette comporte plusieurs étapes à effectuer en parallèle ou en série. Au final, la logique discrète de cet ensemble de recettes devient très complexe. La simulation hybride peut alors se révéler une aide précieuse dans la mise en place de ces recettes. Pour décrire ces recettes plusieurs formalismes peuvent être utilisés : Grafset, machines à états, réseaux de Pétri...

Les simulateurs de systèmes hybrides possèdent des mécanismes de collaboration entre les parties discrète et continue. Pour la partie continue, le simulateur doit être capable d'intégrer efficacement les systèmes algébro-différentiels. Pour la partie discrète, le formalisme utilisé doit permettre de représenter facilement les changements potentiels. De plus, le simulateur doit posséder un calcul automatique de conditions initiales cohérentes et une détection rigoureuse des événements. Cette dernière est primordiale car une erreur dans la séquence de détection des événements peut conduire à des simulations très différentes.

C'est dans ce contexte que le département Procédés des Systèmes Industriels du Laboratoire de Génie Chimique a développé des travaux depuis près de 20 ans sur la simulation dynamique et la simulation dynamique hybride. Le résultat de tous ces travaux a donné naissance à la plateforme de simulation des systèmes dynamiques hybrides : PrODhyS (détaillée dans le chapitre 2). Ce travail de recherche s'inscrit dans la continuité de ces travaux et propose une évolution de cette plateforme. Comme il est mentionné précédemment, la partie optimisation est importante pour la simulation des systèmes dynamiques. L'objectif de ce travail est d'inclure dans la plateforme une méthode robuste d'optimisation dynamique.

Ce manuscrit comporte 4 chapitres. Le premier chapitre, proposera une présentation plus détaillée des systèmes dynamiques hybrides avec les objectifs, les approches et les outils. Le deuxième chapitre se focalise sur l'optimisation des systèmes dynamiques d'un point de vue théorique et sur les méthodes existantes. Le troisième chapitre est centré sur la présentation détaillée de la méthode implantée dans la plateforme ainsi qu'une analyse critique de ses possibilités. Enfin et avant de conclure, le dernier chapitre présente des exemples concrets de mise en application de cette méthode.

CHAPITRE 1

Modélisation et Simulation des Systèmes Dynamiques Hybrides

I Introduction

La plupart des procédés industriels réels évoluent selon des sous processus continus qui sont démarrés, arrêtés par des commandes à états discrets (dont les entrées dépendent des sous processus continus). Par conséquent, les procédés ont rarement un comportement purement discret ou purement continu mais plutôt un mixte des deux. Ces systèmes dynamiques à double composante comportementale (dynamique continue et événementielle) sont nommés : systèmes dynamiques hybrides (SDH).

Comme le précise Zaytoon (2001), la représentation continue ou discrète des modèles est directement liée à la nature des variables d'état et temporelle, qui peuvent être : soit continues, soit discrètes, soit symboliques. C'est à partir de cette classification sur les variables d'état que découlent les modèles des systèmes à dynamique continue ou à événements discrets. Pour les premiers, les variables d'état sont continues et la variable temporelle est soit continue (systèmes constitués d'équations différentielles et algébriques), soit discrète (systèmes échantillonnés). Quant aux modèles des systèmes à événements discrets, les variables d'état sont symboliques (voire discrètes) et la variable temporelle est soit symbolique (événements

discrets non temporisés, seulement occurrence d'événements) soit continue ou discrète (événements discrets temporisés, prise en compte d'information temporelle).

Une modélisation rigoureuse pour les SDH nécessite la prise en compte des interactions entre les aspects continus d'une part et les aspects séquentiels d'autre part. Or culturellement, certaines communautés scientifiques se sont focalisées sur les problèmes à dynamique continue puis ont essayé de prendre en compte les aspects discrets. A contrario, d'autres communautés sont parties des aspects événementiels pour y inclure les éléments continus. Ces façons d'aborder la modélisation des SDH ont conduit à de nombreux formalismes. Le comportement à dynamique continue est régulièrement représenté par un système d'équations différentielles et algébriques tandis que le comportement à dynamique discrète est plutôt traduit par un ensemble d'états et de transitions. Les nombreuses voies adoptées peuvent se classer en trois grandes catégories (Flaus (1998), Zaytoon (2001)) :

- Extension des modèles des systèmes à dynamique continue
- Extension des modèles des systèmes à événements discrets
- Les approches mixtes qui concilient les modèles discret et continu, l'aspect hybride est traité à la jonction des deux parties.

Ce chapitre propose de suivre ce découpage pour arriver à la présentation de la modélisation et de la simulation des systèmes dynamiques hybrides. La première partie traite des modèles à dynamique continue puis de leur extension vers les SDH. La seconde partie est calquée sur la première mais avec comme point de départ les systèmes à événements discrets (pour aller également vers les SDH). Enfin, avant de conclure, la dernière partie s'attache à présenter les approches mixtes.

II Du Continu vers les Systèmes Dynamiques Hybrides

II.1 Les Modèles à Equations différentielles et Algébriques

Dans le domaine du génie des procédés, la partie qui concerne la modélisation des systèmes continus est très importante. L'expression « procédés continus » à une connotation mathématique par rapport au temps. Ces procédés assurent une production permanente en bout d'atelier. Avec les procédés continus est apparue la notion de régime permanent. Le régime permanent implique qu'aucune grandeur (débit, température, pression...) ne varie dans le temps lorsque le point nominal est atteint (ces grandeurs restent constantes). Cette vision est, bien entendu, idéalisée car dans la réalité industrielle, ces grandeurs fluctuent autour de valeurs moyennes. Toutefois, le régime permanent ne couvre pas l'ensemble des problèmes, comme par exemple : le démarrage ou l'arrêt d'installations, la mise en place de systèmes de régulation... Par contre la simulation dynamique permet d'accéder à ce genre d'information.

De manière générale, un système dynamique continu peut être vu comme un système multi variables caractérisé par un vecteur d'état réel de dimension finie. Il suppose un certain niveau de continuité (dérivabilité), de définition pour les variables : leurs évolutions sont continues dans le temps (exemple : la température...). Ainsi un système continu peut être décrit par un ensemble d'équations différentielles ordinaires et/ou algébriques et finalement s'écrire sous la forme implicite :

$$F\left(x^n, x, p', u, t\right) = 0 \quad (1.0)$$

$$x(t_0) = x_0$$

$$x^n(t_0) = x_0^n$$

Avec F : jeu d'équations (différentielles et/ou algébriques)

x : vecteur des inconnues du système

x^n : vecteur des dérivées à l'ordre n par rapport à la variable indépendante (t)

p' : ensemble de paramètres opératoires du système

u : vecteur des paramètres de contrôle

t : variable indépendante (généralement le temps)

Cette représentation générale permet de décrire des phénomènes non linéaires, très présents en génie des procédés. La connaissance au point initial t_0 , du vecteur x et de ses dérivées successives (respectivement x_0, x_0^n), des paramètres opératoires et de contrôle, permet de déterminer l'état ultérieur du système par intégration.

Dans le domaine du génie des procédés, la complexité des systèmes rencontrés implique l'emploi de modèles de représentation basés sur une description détaillée des mécanismes physico-chimiques. La connaissance de ces phénomènes est souvent une étape limitante et la conception des modèles associés constitue une tâche relativement ardue.

II.2 Les Bond Graphs

Introduction

L'outil bond graph (graphe de liaison ou graphe de lien) a été défini par Paynter (1961) puis a été développé par Karnopp et Rosenberg (1996). Les bond graphs constituent un outil graphique permettant de décrire les échanges d'énergie dans le système étudié. Issu des domaines de la mécanique, l'hydraulique et l'électrique, cet outil de modélisation dynamique est applicable à de nombreux domaines de la physique, facilitant ainsi les échanges pluri disciplinaires de part sa nature graphique. Toutefois, malgré quelques articles dans le domaine, les systèmes thermiques, thermodynamiques ou le génie des procédés restent encore de vastes domaines d'exploration en raison de la complexité des phénomènes mis en jeu.

Avec les bond graphs, les phénomènes physiques sont décrits sous forme de graphes plutôt que par la constitution de bibliothèques de modèles basés sur des équations différentielles non linéaires. A l'aide d'un langage universel, les bond graphs permettent d'afficher la nature des échanges de puissance dans le système tels que les phénomènes de stockage, de transformation et de dissipation de l'énergie, tout en mettant en évidence la nature physique et la localisation des variables d'état. De plus, grâce à sa représentation, la modélisation est évolutive (se conférer à l'exemple ci-après) car elle permet d'affiner le modèle par simple ajout de nouvelles branches sur le graphe, sans avoir à reprendre la totalité de la modélisation.

Comme le précise Ould Bouamama et Dauphin (2005), ce langage répond particulièrement bien à plusieurs besoins par les caractéristiques suivantes :

- Une approche énergétique qui permet une décomposition du système étudié en sous systèmes qui échangent de la puissance et qui structurent la procédure de modélisation.
- Une terminologie unifiée pour tous les domaines physiques, fondée sur la notion d'analogie entre phénomènes.
- Une représentation graphique pour visualiser les transferts de puissance, mais aussi de causalité.
- Une souplesse inhérente qui permet de faire évoluer le modèle en ajoutant des phénomènes négligés.
- Une écriture systématique des équations mathématiques issues du modèle bond graph sous forme d'équations différentielles.
- Un support pour une analyse structurelle des propriétés du modèle en vue de la conception de systèmes de surveillance ou de commande.

Principe des Bond graphs

Un bond graph est constitué d'arcs et de nœuds. Ces derniers sont représentés par des caractères alphanumériques et permettent de relier un élément à un des éléments environnant par l'intermédiaire d'un arc. Alors, ces liens traduisent l'échange d'énergie entre deux ports connectés. Le sens de la demi-flèche décrit le sens positif du transfert d'énergie. Ce transfert est calculé grâce à deux variables : un effort (e) et un flux (f) dont le produit donne l'énergie échangée. La puissance instantanée en fonction du temps s'écrit $P=e(t)*f(t)$. D'après le 1^{er} principe de la thermodynamique, qui stipule que l'énergie est conservative, un tel graphe est continu de puissance : le bilan instantané d'énergie est automatiquement satisfait.

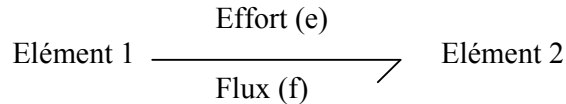


Figure 1.1 : Représentation Bond Graph

A titre d'exemple, le tableau 1.1 présente quelques couples e et f pour des domaines physiques.

Domaine Physique	Effort	Flux
Electrique	Tension (V)	Courant (A)
Mécanique	Force (N)	Vitesse (m/s)
Rotation	Couple (N.m)	Vitesse de rotation (rad/s)
Hydraulique	Pression (Pa)	Débit volumique (m ³ /s)
Thermique	Température (K)	Flux d'entropie (J/(K.s))
Chimique	Potentiel Chimique (J/mol)	Flux molaire (mol/s)
Cinétique Chimique	Affinité Chimique (J/mol)	Vitesse de réaction (mol/s)

Tableau 1.1 : Couple e et f pour différents domaines

Remarque : Pour la description dynamique des systèmes, deux autres variables sont définies : variables d'énergie. Ces variables notées p(t) et q(t) sont calculées par l'intégrale des variables d'effort et de flux et sont respectivement nommées : moment généralisé (impulsion) et déplacement généralisé :

$$p(t) : \text{moment généralisé, } p(t) = \int_0^t e(t)dt + p(0)$$

$$q(t) : \text{déplacement généralisé, } q(t) = \int_0^t f(t)dt + q(0)$$

Remarque : En thermodynamique, les variables dont l'uniformité est nécessaire pour réaliser l'équilibre sont qualifiées d'intensives (température, pression, concentration, potentiel chimique...), d'autres sont extensives (volume, énergie interne, entropie...).

Pour le génie des procédés, il est possible d'avoir une modélisation par bond graph pour laquelle le produit effort flux ne donne pas une puissance : certaines variables sont remplacées par des variables plus faciles à manipuler. On parle alors de pseudo bond graph. C'est le cas pour des variables du type ; potentiel chimique, flux d'entropie, flux molaire qui sont complexes et souvent non mesurables et n'obéissent pas à des lois de conservation simples (entropie n'est pas conservée). Cependant les propriétés classiques des bond graphs sont toujours valables.

Pour réaliser un bond graph, il faut posséder des éléments qui représentent soit des briques élémentaires soit des connexions entre ces briques (tableaux 1.2 et 1.3).

Symbole	Composants	Equations sans causalité
R : r	Résistance, Frottement	$e-r*f=0$
I : i	Inductance, inertie	$e-i *df/dt=0$
C : c	Capacité	$f-c*de/dt=0$
GY	Gyrateur	$e_1=r*f_2 \quad e_2=r*f_1$
TF	Transformateur	$e_1=n*e_2 \quad f_2=n*f_1$
Se	Source d'effort	$e= \text{constante}$
MSe	Source d'effort contrôlée	$e=e(\text{entrée})$
Sf	Source de flux	$f= \text{constante}$
MSf	Source de flux contrôlée	$f=f(\text{entrée})$

Tableau 1.2 : Briques élémentaires

Connexion	Valeurs constantes	Equations
1	Flux	Somme (e_i)=0
0	Effort	Somme (f_i)=0

Tableau 1.3 : Connexions

Remarque : Dans certaine classification, GY et TF sont considérés comme des connexions (tout comme 0 et 1).

Exemple d'application

A titre l'application, l'exemple présenté par Couenne et al (2006a) est brièvement repris dans cette partie : il s'agit du mouvement oscillatoire sans frottement (dans un premier temps) d'un liquide dans un tube en U de diamètre d, figure 1.2.

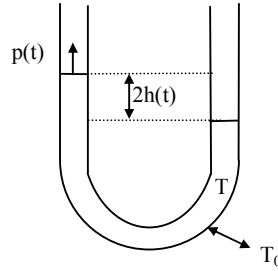


Figure 1.2 : Description du problème

- où p : est la quantité de mouvement par unité de masse
- l : la longueur de la colonne de liquide
- ρ : la masse volumique liquide

Le bond graph de cet exemple est :

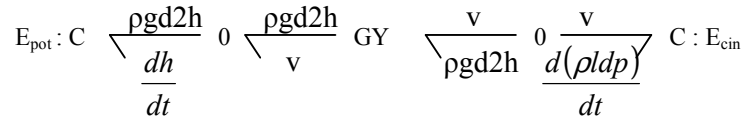


Figure 1.3 : Bond Graph du tube en U

Les éléments C correspondent à de l'accumulation d'énergie, auxquels est associé un couple effort-flux. L'effort décrit l'équilibre tandis que le flux fait l'objet du bilan.

Pour l'énergie cinétique, l'effort est la vitesse et le flux correspond à la quantité de mouvement, par conséquent une force, contrairement à l'énergie potentielle, où la vitesse joue le rôle de flux et la force devient la variable d'effort. Le gyrateur GY permet de décrire cet échange conservatif du rôle des variables.

Les bilans instantanés sur les flux sont représentés par les briques 0, où la somme des flux est nul (effort commun), tableau 1.3.

Au final :

Bilan de quantité de mouvement (Energie Cinétique) : $\frac{d(\rho l d p)}{dt} = -\rho g d 2 h$

Bilan d'espace (Energie potentielle) : $\frac{dh}{dt} = v$

Compte tenu des hypothèses, cet exemple est un oscillateur harmonique tel que les deux domaines énergétiques (potentiel et cinétique) échangent de l'énergie sans perte. Par conséquent l'énergie totale (E_{tot}) est constante. Le bilan instantané d'énergie s'exprime alors :

$$\frac{dE_{tot}}{dt} = v \frac{d(\rho l d p)}{dt} + \rho g d 2 h \frac{dh}{dt} = v(-\rho g d 2 h) + (\rho g d 2 h)v = 0$$

Il reflète bien la propriété du graphe : la continuité en puissance.

Si le fluide n'est plus parfait mais visqueux de température uniforme T (T_0 est la température extérieure). Les frottements conduisent à une augmentation de température du liquide et un échange de chaleur se crée avec le milieu extérieur. La prise en compte de cette nouvelle situation se traduit par l'ajout au bond graph d'une brique C associée à l'énergie interne du liquide. Pour modéliser ce nouveau problème, le bond graph précédent reste toujours valable, il suffit de l'étendre en ajoutant la partie échange de chaleur au niveau de la connexion 0, du côté énergie cinétique. L'annexe 1 donne une description détaillée de ce cas.

La causalité

Une des propriétés importantes des bond graphs est la causalité qui régit la relation énergétique attachée à un lien. Cette représentation cause effet améliore la compréhension des systèmes. La causalité est matérialisée par une barre à une extrémité de la demi-flèche.

Soit deux éléments A et B qui échangent de l'énergie, deux cas de figure sont possibles :

- A applique un effort à B et A réagit en envoyant un flux (figure 1.4a)
- A applique un flux à B et B renvoie un effort (figure 1.4b)



Figure 1.4 : Représentation de la causalité.

Le trait est placé au plus près de l'élément pour lequel l'effort est une donné.

Attention : il ne faut pas confondre le sens de la demi-flèche qui traduit le sens de transfert de l'énergie (choisi arbitrairement) et le trait de causalité qui est fixé par les éléments et leurs connexions.

Conclusion

Cette partie permet de décrire succinctement les caractéristiques principales des bond graphs. Ce langage graphique universel permet d'améliorer la représentation et la gestion des modèles. Tout d'abord, cette « norme » graphique pluridisciplinaire reste lisible par une large communauté (même des personnes hors du domaine de compétence du système modélisé peuvent comprendre) car ils modélisent directement les phénomènes physiques mis en œuvre. De plus, les possibilités d'amélioration, d'extension, des modèles sont faciles à mettre en œuvre comme le démontre l'exemple. Comme le souligne Couenne et al (2006a), les difficultés liées à l'évolution des outils logiciels que rencontrent les ingénieurs de

modélisation pour la gestion des modèles risquent de continuer à exister, même si les bond graphs sont utilisés. Toutefois, elles seront diminuées grâce au gain de clarté de la documentation des modèles.

Les bond graphs commencent à faire leur apparition dans le domaine du génie des procédés comme par exemple Couenne et al (2006b) pour un réacteur chimique, Ceynowa et Adanczak (2001) pour un réacteur à membranes, Simon et al (1996) pour une séparation par membrane.

II.3 Extension des modèles continus

Dans le cadre de l'extension des modèles continus vers les systèmes dynamiques hybrides (SDH), une première approche proposée repose sur l'utilisation de variables mixtes. Celle-ci consiste à introduire des variables booléennes ou entières au sein du modèle continu. Par exemple, l'état « ouvert » ou « fermé » d'une vanne tout ou rien peut être représenté par une variable booléenne égale respectivement à un ou à zéro. Cette approche peut également être illustrée au travers de l'exemple de l'évaporation de Rayleigh, qui consiste à chauffer un mélange liquide jusqu'à son point de bulle afin de le vaporiser.

Pour illustrer ceci, considérons une cuve, contenant initialement un binaire ($n_c = 2$) de composition x , de rétention liquide U_l (la rétention vapeur est supposée négligeable devant la rétention liquide) et recevant une quantité de chaleur Q . Au point de bulle, une phase vapeur de composition y apparaît et s'échappe dans l'atmosphère avec un débit v . Le modèle continu global associé à ce système est le suivant :

$$\begin{aligned} \text{Bilan matière global :} & \quad \frac{dU_l}{dt} + V = 0 \\ \text{Bilans matière partiels :} & \quad \frac{d(U_l \cdot x_i)}{dt} + V \cdot y_i = 0 \quad (i = 1, n_c) \\ \text{Bilan d'énergie :} & \quad \frac{d(U_l \cdot h)}{dt} + V \cdot H - Q = 0 \\ \text{Relations d'équilibre :} & \quad y_i - K_i \cdot x_i = 0, \quad (i = 1, n_c) \\ \text{Somme des fractions molaires :} & \quad \sum_i x_i - \sum_i y_i - \tau = 0 \\ \text{Equation de switch :} & \quad v \cdot \tau = 0 \end{aligned}$$

La gestion de la discontinuité liée au passage de l'état liquide à l'état liquide / vapeur est gérée grâce à l'introduction de la variable booléenne τ et de l'équation de « switch » associée. Initialement, le système est à l'état liquide et la variable τ est égale à 1. L'équation de switch garantit alors que le débit vapeur est égal à 0. Lorsque la température de bulle est atteinte, la variable τ change de valeur et devient égale à 0, entraînant alors la prise en compte de la phase vapeur dans le modèle.

Notons cependant que lorsque le nombre de variables booléennes augmente, cette approche peut rapidement devenir complexe.

Une autre extension consiste à introduire des éléments de commutation dans le modèle continu. Cela donne naissance à des graphes d'états représentant les modes de fonctionnement du système. Ils sont associés d'une part, à des systèmes d'équations différentielles modélisant le comportement de la partie continue, et d'autre part, à des fonctions de saut traduisant les discontinuités lors des changements de modes.

Justement, une voie d'amélioration des bond graphs, présentés précédemment, est de leur introduire des fonctions de saut. Ils sont alors nommés bond graph à commutations (Zaytoon 2001, Buisson et Cormerais, 1998 ...). Les éléments de commutations constituent l'interface de la partie événementielle vers la partie continue. Plusieurs modélisations sont possibles comme par exemple :

- L'utilisation de résistance variable (jonction R) passant instantanément, à la commutation, d'une valeur r_0 à une valeur r_f .
- Le passage d'un état à un autre peut être fait au niveau d'un transformateur (TF) en lui associant une résistance (m) dont la valeur change à la commutation.
- ...

III Des événements discrets vers les Systèmes Dynamiques Hybrides

Les systèmes à événements discrets (SED) englobent de nombreux domaines d'applications : production manufacturière, les transports, les réseaux de télécommunications, l'automatique etc... De manière générale, un SED peut être vu comme un système dynamique dont l'espace continu a été discrétisé en un espace d'états discrets. Pour ces systèmes, la transition permettant le passage d'un état discret à un autre dépend de l'occurrence d'événements. Ces évolutions, nommées trajectoires, sont basées sur une succession d'états et de transitions.

Couramment, pour l'étude de ces problèmes, le comportement dynamique d'un SED ne s'attache qu'à l'ordre d'occurrence des événements et ne tient pas compte explicitement du temps. Ces systèmes sont des SED non temporisés. Par exemple sur la figure 1.5 la trajectoire est décrite par la séquence : A, B, C, D.

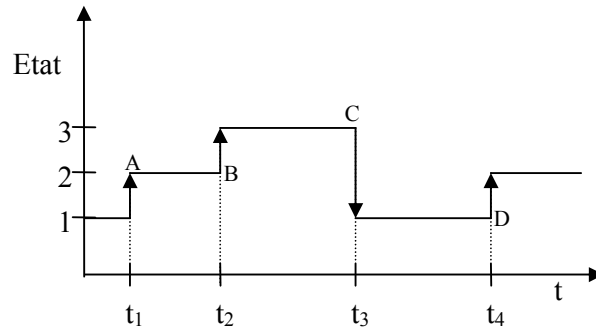


Figure 1.5 : Exemple de trajectoire pour un SED

Toutefois, dans certaines applications, la prise en compte du temps est primordiale et il est intégré aux modèles de façon à spécifier les trajectoires d'événements et leurs temps. Dans ce cas les SED sont qualifiés de temporisés. Les SED temporisés constituent un premier pas vers l'intégration de phénomènes hybrides dans le système.

Le but de cette partie est de présenter les modèles définis pour représenter les SED.

III.1 Les Automates à états finis

Les automates à états finis (Hopcroft et Ulman 1979) constituent le formalisme de base des SED. Le modèle décrit un système possédant un nombre d'états qui est fini. Ce type d'automates est souvent décrit par un quadruplet. Tout d'abord, trois de ces éléments principaux sont représentés sous forme d'un graphe sur lequel les états correspondent aux nœuds, les arcs aux transitions qui relient ces nœuds et les étiquettes placées sur les arcs aux événements qui déclenchent ces transitions. Enfin, le dernier élément du quadruplet donne l'état initial.

A titre d'illustration, l'exemple proposé par Zaytoon (2001) est repris. La figure 1.6 présente le modèle d'une machine simple à 3 états : Arrêt (Ar), Marche (Ma), Panne (Pa). L'état initial Ar est désigné par une flèche entrant. Dans cet exemple, il y a 4 transitions associées chacune à l'un des 4 événements.

Si la machine précédente est dupliquée pour former un atelier à deux machines indépendantes évoluant en parallèle. L'automate est alors représenté par 9 états et 24 transitions. Si l'atelier comporte 3 machines, l'automate sera constitué de 27 états et 108 transitions et ainsi de suite. Ce petit exemple met en évidence l'inconvénient de ce modèle qui réside dans l'explosion combinatoire du nombre d'états accessibles lorsque le système à modéliser devient trop complexe ou comporte de nombreuses opérations pouvant s'exécuter en parallèle. De plus, les phénomènes tels que l'exclusion mutuelle ou le parallélisme ne sont pas représentés explicitement, ce qui réduit la lisibilité du modèle. Afin de maîtriser l'explosion combinatoire, il faut avoir recours à d'autres outils de modélisation basés sur des

mécanismes tels que l'abstraction, la modularité, l'agrégation et la décomposition hiérarchique.

- A : début du travail
- B : Fin du travail
- C : Panne de la machine
- D : Réparation de la machine

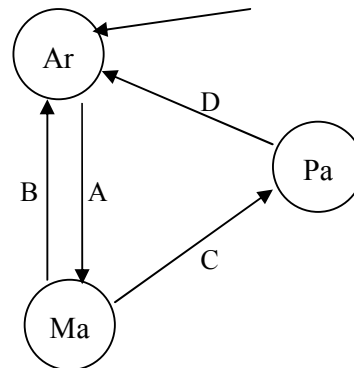


Figure 1.6 : Exemple d'automate à états finis

III.2 Le Grafcet

Le grafcet (AFNOR, 1982) est un outil normalisé sur le plan international depuis 1988 (CEI, 1988). Il est représenté par un graphe cyclique, composé alternativement d'étapes (états du système) et de transitions (changements d'état), ces éléments étant reliés entre eux par des arcs. Son avantage majeur est qu'il exprime efficacement les phénomènes de synchronisation et de parallélisme. Il constitue également un outil très bien adapté aux systèmes automatisés. Toujours en reprenant l'exemple de Zaytoon (2001), la figure 1.7a illustre deux grafcets décrivant l'évolution en parallèle de deux machines indépendantes et identiques (similaires à celle de la figure précédente).

La figure 1.7b met en évidence le parallélisme, ou dans cette modélisation, le démarrage M est donné simultanément aux deux machines.

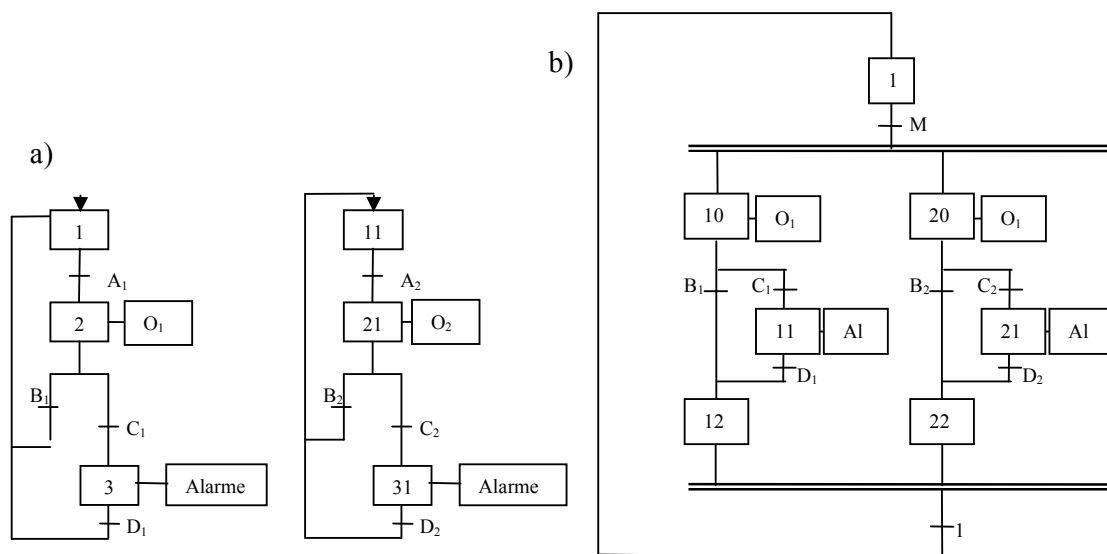


Figure 1.7 : Illustration de Grafcet.

Pour ces deux choix de modélisation, l'ajout d'une machine se traduit par un grafcet partiel ou une séquence parallèle. Au final, le nombre d'étapes et de transition croît linéairement avec l'ajout de machine parallèle et non pas exponentiellement comme dans le cas précédent. Toutefois, le nombre d'états reste identique à celui de l'automate à états finis.

III.3 Les Réseaux de Petri

Introduits par Carl Adam Petri en 1962, les réseaux de Petri (RdP) constituent un outil de représentation formel, graphique et mathématique très bien adapté pour la modélisation, l'analyse et le contrôle des SED. Ils disposent en effet d'une définition formelle et de nombreuses techniques de vérification automatique des propriétés (borné, vivant, réinitialisable). Ils permettent de décrire efficacement les contraintes de synchronisation, de parallélisme et de séquençement.

Graphiquement, un réseau de Petri est un graphe orienté composé de deux types de nœuds : les places (cercles) et les transitions (traits). Des arcs orientés, auxquels un poids (nombre entier strictement positif) est éventuellement associé, relie les places aux transitions. Le graphe constitue la partie statique du réseau. L'aspect dynamique est introduit par les jetons qui se déplacent sur le réseau. Leur position sur les places à un instant donné est représentée par un vecteur et définit le marquage du réseau. Lorsque le poids n'est pas signalé, il est par défaut égal à un.

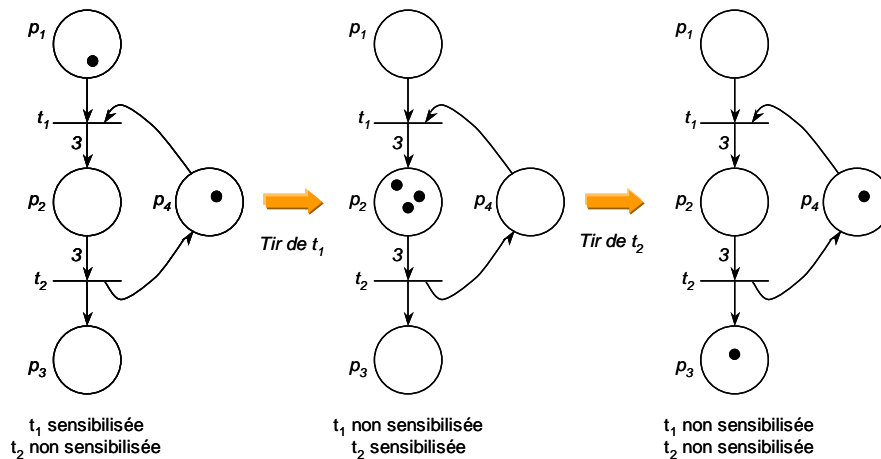


Figure 1.8 : Franchissement d'une transition

Les jetons se déplacent sur le réseau de Petri selon des règles d'évolution. Une transition t_j est sensibilisée si chaque place d'entrée p_i (place située en amont) contient au moins k_{ij} jetons, k_{ij} étant le poids associé à l'arc (p_i, t_j) reliant la place p_i à la transition t_j . Le franchissement de la transition consiste alors à retirer k_{ij} jetons de chaque place d'entrée p_i et à ajouter k_{ji} jetons (k_{ji} étant le poids associé à l'arc (t_j, p_i) reliant la transition t_j à la place p_i) à chaque place de sortie de la transition t_j (place située en aval). La figure 1.8 représente le tir des transitions t_1 et t_2 et l'évolution du marquage qui en résulte.

Formellement, un réseau de Petri discret autonome initialement marqué est un quintuplet $N = \langle P, T, Pre, Post, M_0 \rangle$ où :

- $P = \{P_1, P_2, \dots, P_{n_p}\}$ est un ensemble fini de places ; $\text{card}(P) = n_p$;
- $T = \{T_1, T_2, \dots, T_{n_t}\}$ est un ensemble fini de transitions ; $\text{card}(T) = n_t$;
- $Pre : P \times T \rightarrow \mathbb{N}$ est l'application d'incidence avant qui définit les arcs entrants des transitions ; \mathbb{N} est l'ensemble des entiers non négatifs ;
- $Post : P \times T \rightarrow \mathbb{N}$ est l'application d'incidence arrière qui définit les arcs sortants des transitions ;
- M_0 est le marquage initial.

Afin d'illustrer le formalisme réseau de Petri, la figure 1.9 représente le fonctionnement de deux machines identiques à celle de la figure 1.6 avec un stock de capacité 5 (Zaytoon, 2001). La machine M_1 produit une pièce pour le stock s'il n'est pas plein et la machine M_2 prélève une pièce dans ce dernier s'il n'est pas vide.

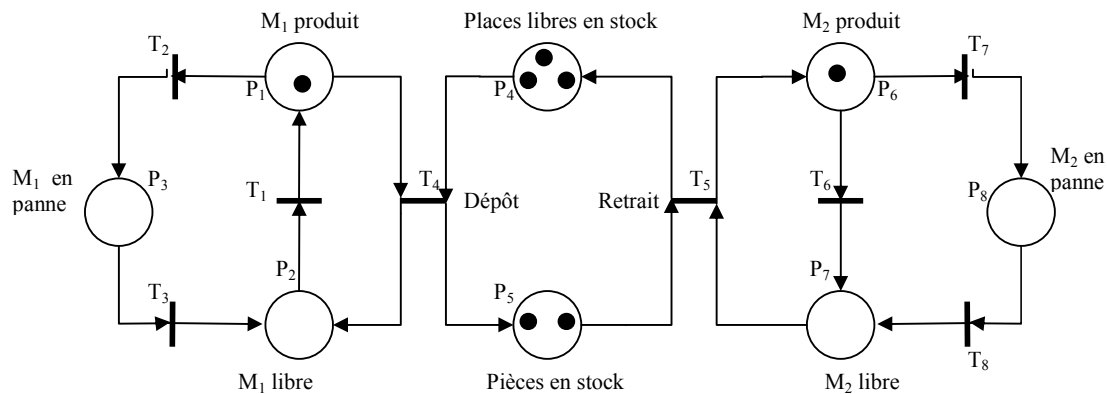


Figure 1.9 : Exemple de réseau de Petri

Même si les réseaux de Petri permettent de gérer efficacement le parallélisme (contrairement aux automates à états finis), le nombre d'états atteignables peut rapidement augmenter la combinatoire du modèle. Pour cette raison, il est apparu nécessaire de définir des extensions permettant de compacter et d'organiser l'information contenue dans le modèle. Dans ce contexte, des modèles de haut-niveau ont été créés ; dans la catégorie des réseaux de Petri (Jensen et Rozenberg, 1991), le premier à faire son apparition fut le réseau de Petri prédicats-transitions.

III.4 Les Statecharts

Les statecharts (Harel, 1987) fournissent une représentation graphique et hiérarchisée du système et le décrivent en termes d'états, de transitions, et d'événements. Ils sont caractérisés par trois grandeurs :

- La profondeur : elle rend la structure hiérarchique en décomposant des supers états en sous états (contenus à l'intérieur du super état). Elle permet de réaliser une agrégation ou une décomposition. La figure 1.10a (Zayton (2001)) présente un modèle affiné de la machine présentée sur la figure 1.6. Cette machine Ma est décomposée en trois sous états : avec pour débiter l'activation du sous état m1, suivi d'une alternance entre m1 et m2 jusqu'à l'occurrence de l'événement g qui active m3. La transition C est sur Ma de façon à spécifier que l'occurrence de cet événement peut provenir de n'importe quel sous état de Ma (évitant ainsi trois transitions C issues des sous états de Ma).
- L'orthogonalité : elle permet de décomposer un état en sous états actifs simultanément (séparés par une ligne pointillée).
- La communication : elle permet d'émettre des événements en diffusion et donc disponibles partout au même moment.

Ces deux dernières notions sont bien illustrées sur la figure 1.10b (Zaytoon (2001)) qui représente le cas de deux machines synchronisées par un stock à deux états : libre (l) ou occupé (o).

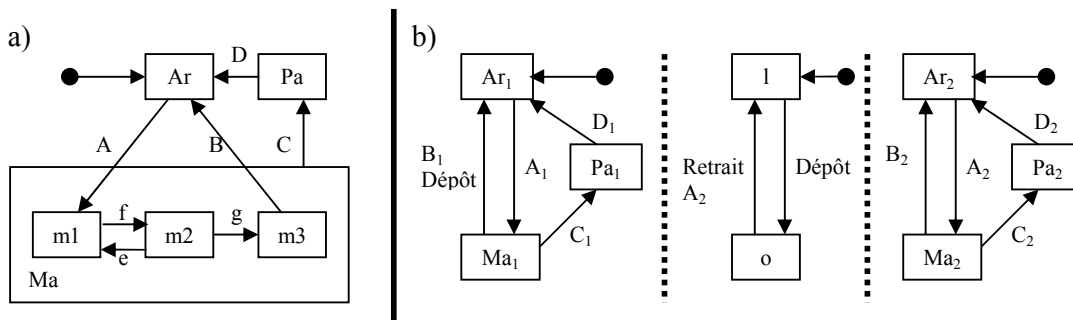


Figure 1.10 : Illustration de statecharts

L'événement de dépôt B₁ indique la fin de l'opération sur Ma₁ (et donc sur la machine 1) et elle entraîne la diffusion de l'action de dépôt, qui est simultanément utilisé au niveau du stock en tant qu'événement pour activer l'état o. De même le retrait du stock (événement retrait sur la transition sortante du sous état o), entraîne la diffusion de l'action A₂ qui est utilisée en tant qu'événement pour la transition d'activation de Ma₂.

Dans les modèles à événements discrets proposés dans les parties précédentes, le temps est pris en compte de manière qualitative. Des extensions de ces formalismes ont été proposées afin de tenir compte des effets qualitatifs du temps et ainsi se rapprocher des systèmes dynamiques hybrides.

III.5 Extension des Modèles à Événements Discrets

Les automates temporisés

Les automates temporisés (Alur et Dill (1994)) constituent une extension des automates à états finis. Ils sont étendus par l'adjonction d'un ensemble d'horloges qui permettent de mesurer et de contrôler le temps écoulé entre les actions. Des contraintes sur les horloges peuvent être associées aux états et aux transitions. Le cas présenté pour les automates à états finis est repris et étendu à titre d'exemple (Zaytoon (2001)) :

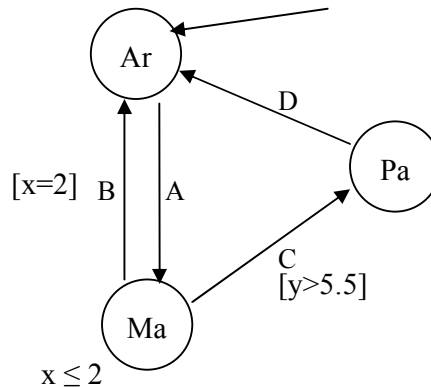


Figure 1.11 : Automates temporisés

Deux horloges ont été ajoutées pour indiquer que : la durée de l'état de marche sur la machine est de 2 unités de temps (UT) (horloge x) et une panne ne peut avoir lieu que 5.5 UT après la dernière réparation (horloge y). Ainsi les éléments suivants permettent à l'horloge x de forcer le retour au sommet Ar après un séjour d'exactly 2 UT dans le sommet Ma si une panne ne survient pas :

- Remise à 0 de x par transition entrant dans Ma
- Impossibilité de rester dans Ma au-delà de 2 UT
- Autorisation de l'exécution de la transition de Ma vers Ar, lorsque $x=2$

L'horloge y est remise à 0 après chaque réparation pour autoriser l'occurrence ultérieure d'une panne après un séjour d'au moins 5.5 UT dans les sommets Ar et Ma.

Extension des Réseaux de Petri

La prise en considération des aspects qualitatifs du temps dans les réseaux de Petri (RdP) a donné naissance aux réseaux de Petri temporisés, temporels et stochastiques.

Réseaux de Petri temporisés :

Dans cette extension des RdP, une durée de franchissement est associée soit à la place (RdP p-temporisé) soit à la transition (RdP t-temporisé). Par conséquent le jeton est indisponible pendant toute la durée de la temporisation, il est donc réservé. Si ce modèle permet d'intégrer les contraintes temporelles fixées a priori, il ne peut pas en revanche tenir compte des variables du système. En effet, si au cours de la simulation une perturbation a lieu alors les temporisations devraient évoluer en conséquence. Or ce n'est pas le cas car les perturbations ne sont pas détectables pendant la simulation.

Exemple : Lors du transfert d'une quantité de produit, une brusque diminution de débit aurait pour conséquence d'augmenter le temps de transfert mais ce n'est pas observé.

Cette remarque met en évidence la nécessité de prendre en compte non seulement le temps mais aussi les variables d'état du système.

Réseaux de Petri temporels :

Les RdP temporels étendent les RdP ordinaires en associant à la transition t , un intervalle fermé $[\theta_{smin}, \theta_s, \theta_{smax}]$. La transition n'est franchissable que lorsque la durée de sensibilisation θ_s est comprise dans l'intervalle. Contrairement aux RdP temporisés, les jetons marquant les places d'entrée de la transition sont disponibles et peuvent éventuellement être engagés dans une autre transition en conflit avec la transition t .

Réseaux de Petri stochastiques

Dans le cas précédent, les durées sont associées aux places et aux transitions du réseau. Ces modèles sont bien adaptés lorsque les aspects quantitatifs sont bien définis. En revanche, certains phénomènes sont imprévisibles (des perturbations comme dans l'exemple ci-dessus) et ne peuvent plus être traités avec ce type de modèles. Les réseaux de Petri stochastiques ont justement été introduits pour modéliser efficacement ces problèmes aléatoires. Ainsi dans ces RdP, les durées de sensibilisation associées aux transitions, résultent d'un tirage aléatoire (l'hypothèse couramment employée est une distribution des durées selon des lois exponentielles).

Autres extensions des Réseaux de Petri

La section précédente présente une liste d'extensions intégrant les aspects continus dans un formalisme initialement dédié aux systèmes à événements discrets. Toutefois, pour avoir les approches les plus couramment citées dans la littérature il faut en ajouter d'autres :

Les Réseaux de Petri continus :

Dans les RdP classiques, le marquage d'une place est discret ce qui génère souvent l'énumération d'un grand nombre d'états accessibles pour un système complexe. A des fins d'améliorations du modèle, le RdP continu considère le franchissement des transitions comme un processus continu (extension du marquage). Le franchissement n'est plus instantané, la marque (remplace le terme de jeton réservé au marquage discret) traverse la transition par quantité infinitésimale : vitesses de franchissement associées aux transitions. Des modèles différents de RdP continus découlent de la nature de ces vitesses : RdP à vitesse constante, à vitesse variable, continue asymptotique.

Les Réseaux de Petri Hybrides

Dans certains cas, la représentation au sein d'un même modèle des aspects continu et discret sont nécessaires car ni le formalisme précédent, ni le formalisme RdP discret ne permettrait de modéliser de tels systèmes. Cela a conduit à la définition des RdP hybrides où coexistent des places et des transitions continues et discrètes.

Les Réseaux de Petri lots

Ces réseaux étendent les précédents pour modéliser les systèmes de production pour lesquels coexistent simultanément des phénomènes d'accumulation et de retard de propagation. Par exemple, la disponibilité d'un lot de produit qui entre sur un convoyeur n'est pas immédiate, il y a un temps de transit de la partie amont vers l'aval (retard de propagation). Si la totalité du flux sortant du convoyeur n'est pas absorbée par la machine située en aval alors une partie du lot peut s'accumuler. Un jeu de paramètres décrivant le lot (densité, longueur) et une loi continue d'évolution temporelle du système active pendant la durée du marquage, sont attribués à la place. De plus des vitesses de franchissement (liées à des informations des places situées en amont et en aval) sont associées aux transitions.

Cette liste est loin d'être exhaustive car de nombreuses dérivations découlent des RdP, notons plus particulièrement des travaux effectués au sein de notre équipe sur les RdP différentiels à objets (Perret (2003)).

IV Modélisation des Systèmes Dynamiques Hybrides

Comme énoncé dans l'introduction de ce chapitre, les différentes approches pour modéliser les SDH se classent en trois grandes parties. Les deux premières (issues des extensions des modèles des systèmes à dynamique continue et à événements discrets ont été brièvement abordées dans les parties précédentes. Il convient maintenant d'aborder la troisième partie à savoir les approches mixtes.

IV.1 Les Modèles Mixtes

Dans une approche mixte, chacune des deux composantes (discrète et continue) est représentée de façon rigoureuse et explicite selon un des formalismes décrits dans les sections précédentes et leur collaboration est prise en compte dans l'interface qui les relie. La résolution du modèle continu déclenche l'évolution des variables au cours du temps et valide certaines transitions. L'évolution du modèle discret engendre alors la mise en place d'un nouvel état discret qui se traduit par l'élaboration d'un nouveau système d'équations. Plusieurs modèles mixtes ont été définis. Cette section se propose d'en présenter quelques-uns.

Les automates hybrides

Les automates hybrides (Alur et al. (1995)) sont une extension des automates à états finis. Ils représentent des systèmes qui intègrent deux composantes : celle ayant un comportement discret, modélisée naturellement par un automate à états finis et celle dont le comportement varie de manière continue dans le temps, modélisée par un système algébro-différentiel. Notons qu'un automate temporisé est un cas particulier d'un automate hybride pour lequel le temps constitue la seule variable continue. Les problèmes soulevés par cette approche résident toujours dans le manque de structuration des données et de lisibilité du modèle.

Les statecharts hybrides

Les statecharts hybrides (Kesten et Pnueli (1992)), qui constituent une extension des statecharts, associent un ensemble d'équations différentielles aux états. Cette représentation renforce la puissance du modèle liée à la structuration des modèles : en effet, l'utilisation de la hiérarchie des états permet de structurer la partie événementielle et l'association d'équations aux états permet de structurer les relations entre les grandeurs continues.

Les réseaux de Petri mixtes

L'approche basée sur les réseaux de Petri mixtes (RdPM) (Valentin-Roubinet, 1999) consiste à interpréter les RdP par des équations différentielles. Un RdP mixte est constitué (Zaytoon, 2001) :

- d'un ensemble de RdP interprétés (David et Alla, 1994 ; Daubas et al., 1994) synchronisés par des événements internes. Dans cette représentation, la place peut être interprétée soit comme une activité (un système algébro-différentiel lui est alors associé et le nombre de jetons qu'elle peut accepter est limité à un, afin de garantir la cohérence du modèle), soit comme une condition logique, une ressource disponible ou l'état d'un stock (aucune équation ne lui est attribuée et le nombre de jetons est illimité) ;

- d'une partie continue qui comprend l'ensemble des variables continues (globales et accessibles de partout dans le modèle et à tout instant) et l'ensemble des équations ;
- d'une interface hybride qui assure la collaboration entre la dynamique continue et la dynamique événementielle. Elle est caractérisée par :
 - un ensemble d'équations actives pour chaque place marquée ;
 - un ensemble d'équations résultant de la concaténation des équations actives de toutes les places marquées des réseaux définies pour chaque configuration de marquage ;
 - une application qui associe à chaque transition, un vecteur dont chaque composante correspond à la fonction de saut sur chacune des variables ;
 - une application qui associe à chaque transition, une condition sur les variables continues devant être vérifiée pour la franchir.

Cette approche a notamment été utilisée dans le cadre de l'ordonnancement. Un cas industriel est traité dans (Pagès et Pingaud (1995)) et présente un outil d'aide à l'ordonnancement d'ateliers batchs à production continue. De même, Valentin-Roubinet (1998) utilise les RdP mixtes pour modéliser une unité de stockage de gaz. Cependant, notons que l'inconvénient majeur des réseaux de Petri mixtes réside dans l'utilisation de variables globales, ce qui ne contribue aucunement à la structuration des modèles.

Les réseaux de Petri prédicats-transitions différentiels

Les réseaux de Petri prédicats-transitions différentiels (RdPPTD) s'appuient sur le formalisme réseau de Petri mixte (RdPM) présenté précédemment et sur les réseaux de Petri prédicats-transitions (RdPPT).

En effet, ils sont nés avec l'idée de tirer avantage des aspects « structuration des données » propres aux RdPPT d'une part et de la représentation rigoureuse de la partie continue propre aux RdPM d'autre part. Dans ce contexte, un RdPPTD représente un système dont le comportement discret est décrit par des réseaux de Petri prédicats-transitions et le comportement continu par des systèmes d'équations différentielles algébriques (Champagnat (1998) ; Champagnat et al. (1998a) (1999)). Dans cette représentation, le comportement hybride est pris en compte d'une part, en associant les variables continues aux jetons et d'autre part, en associant un système algèbro-différentiel aux places permettant de faire évoluer les variables d'état continues. Un jeton mis dans une place déclenche la résolution du système différentiel correspondant et provoque ainsi l'évolution continue au cours du temps des variables d'état. Ceci constitue une différence fondamentale avec le modèle RdPPT dans lequel les attributs des jetons ne sont modifiés que lors du franchissement des transitions.

Un exemple d'application concernant un processus continu de stockage de gaz est traité dans (Champagnat et al. (1998b)).

IV.2 Simulation des Systèmes Dynamiques Hybrides

La simulation reste un outil privilégié lorsqu'il s'agit, par exemple, d'améliorer la conception et les performances sans recourir systématiquement à des tests réels sur l'unité en fonctionnement (procédures coûteuses et très hasardeuses), de mettre en évidence et d'éliminer les défauts conceptuels, d'augmenter le degré de fiabilité d'un schéma en permettant de tester les plages de fonctionnement normales ou anormales de façon identique à ce qu'autoriserait la construction d'un pilote. Cette section est consacrée à la simulation des SDH. Dans ce contexte, nous présenterons, dans un premier temps, l'architecture générale d'un noyau de simulation pour les SDH.

Architecture générale d'un simulateur dynamique hybride

L'architecture générale d'un simulateur dynamique hybride est décrite sur la figure 1.12.

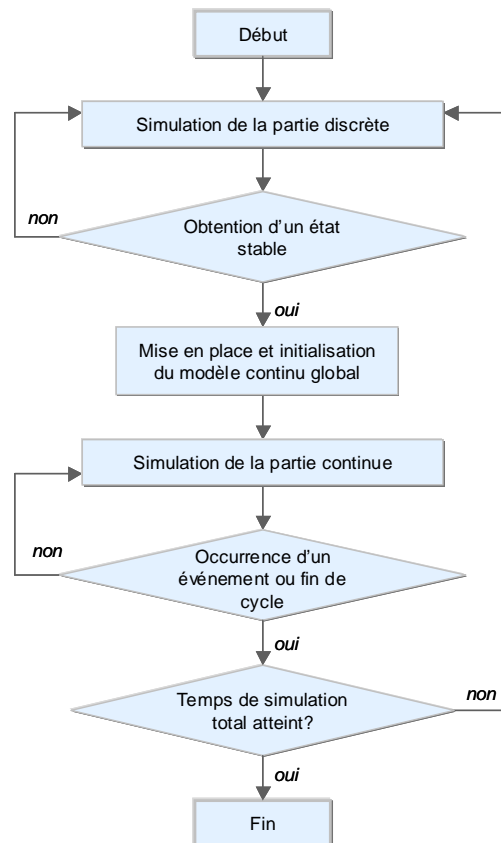


Figure 1.12 : Architecture Générale

La simulation dynamique hybride intègre plusieurs étapes : la simulation de la partie discrète, la simulation de la partie continue et la mise en place du modèle continu global.

Simulation des parties discrètes et continues

La simulation de la partie discrète consiste à jouer le modèle discret : les marques se déplacent sur le graphe en franchissant les transitions jusqu'à l'obtention d'un état stable (c'est à dire jusqu'à ce que plus aucune transition ne puisse être franchie).

La simulation de la partie continue consiste à résoudre un système d'équations composé traditionnellement d'équations différentielles ordinaires et partielles et d'équations différentielles algébriques.

Plusieurs méthodes de résolution existent : les méthodes multi-pas dont la méthode BDF (Backward Differentiation Formulas) et la méthode de Gear (1971) (méthode implicite), et les méthodes à un pas comme la méthode IRK (Runge Kutta Implicite) qui est particulièrement efficace dans le cas des systèmes caractérisés par de fréquentes discontinuités (Brenan et al., 1989).

Notons par ailleurs que pour pouvoir utiliser ces méthodes en simulation continue, il doit exister un mécanisme permettant de stopper l'intégration lorsqu'un événement survient. Deux types d'événements existent : les événements temporels pour lesquels la date d'occurrence est prédéfinie et les événements d'état pour lesquels la date d'occurrence n'est pas connue a priori puisqu'ils se déclenchent lorsque la valeur d'une variable d'état franchit un seuil donné. Une fois l'événement détecté, il est alors nécessaire de le localiser temporellement de manière précise.

Mise en place du modèle continu global

Le modèle continu global est obtenu par concaténation des sous-modèles actifs du système. La structure qui le caractérise est par conséquent souvent variable, puisque le nombre d'équations et le nombre de variables sont amenés à changer au cours de la simulation. Les intégrateurs doivent alors être dotés d'algorithmes de vérification et d'analyse du système d'équations afin de réduire l'index du problème, de vérifier la cohérence du modèle (nombres d'équations et de variables égaux), de permettre l'initialisation ou la réinitialisation du modèle....

IV.3 Classification des outils de simulation

Devant la complexité grandissante des procédés, l'IPAO (Ingénierie des Procédés Assistée par Ordinateurs) s'est fortement développée et a donné naissance à de nombreux progiciels, intégrant les fonctions de conception, d'exploitation et de conduite.

Ainsi, face à un problème de modélisation, de simulation ou d'optimisation et selon l'acteur concerné, plusieurs types d'outils sont actuellement proposés (Moysse, 2000), figure 1.13.

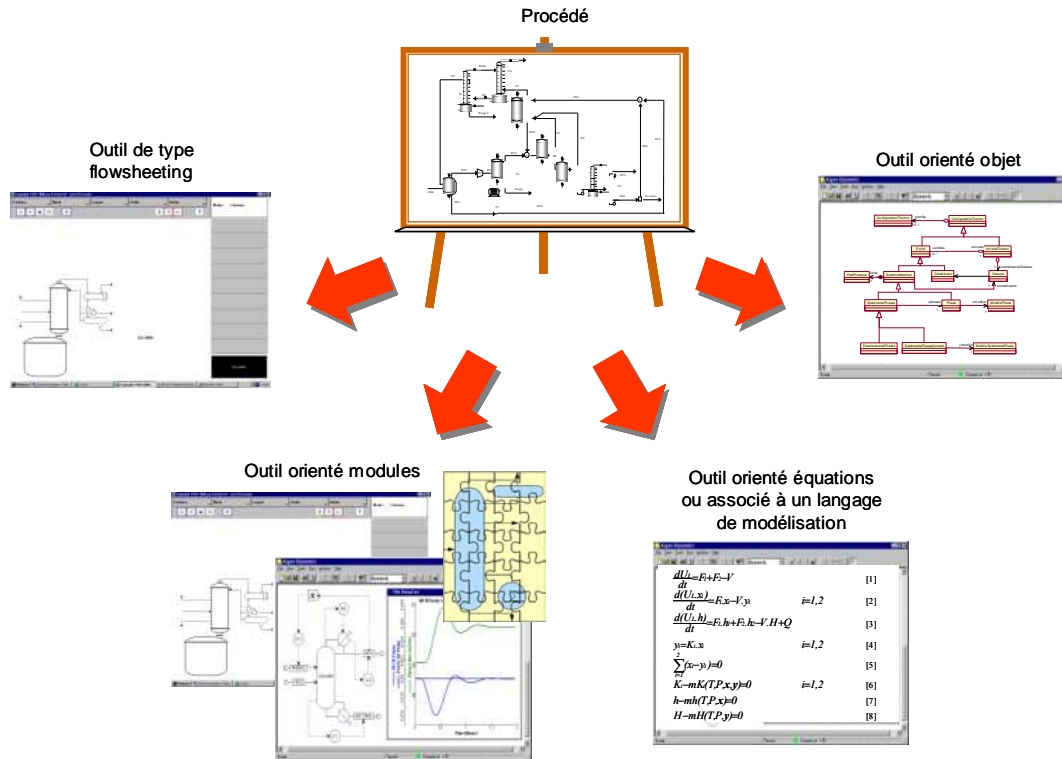


Figure 1.13 : Outils de simulation

Un premier type d'outil rencontré est basé sur une approche de type « flowsheeting ». L'ingénieur construit sans programmer son procédé en connectant des appareils prédéfinis regroupés dans des bibliothèques dédiées au génie des procédés. Son utilisation se fait soit au travers d'une interface graphique de type « drag and drop » (« glisser et poser »), soit par un langage dédié, non extensible. Chaque procédé est représenté par des blocs prédéfinis connectés entre eux pour des échanges d'énergie, de matière et d'information.

Avec l'approche « flowsheeting », l'ingénieur dispose ainsi d'un environnement de simulation dont le but est d'induire un travail d'implémentation réduit grâce à l'intégration de l'ensemble des outils de modélisation, de simulation et d'analyse au sein de la même structure. Cependant, un environnement intégré complet qui répond à tous les besoins n'existe pas, tant la complexité et l'éventail des systèmes et des savoir-faire requis est vaste. Par ailleurs, dans cette approche, l'acteur se heurte à des systèmes propriétaires souvent fermés et monolithiques, offrant peu de possibilités d'extension durant l'analyse de systèmes spécifiques. En effet, les modèles sont pré-codés et l'acteur ne peut utiliser que les appareils proposés dans la bibliothèque. Pour pallier cette limitation, un groupe d'industriels, de fournisseurs de logiciels et d'universitaires se sont regroupés pour définir un standard international CAPE-OPEN (Belaud et Pons, 2002 ; Gani et Braunschweig, 2002). En effet, le projet CAPE-OPEN vise à standardiser une interface pour favoriser l'interopérabilité des composants et par conséquent offrir des solutions logicielles plus flexibles et plus ouvertes.

On parle alors d'outils reposant sur une approche orientée modules. Ce projet collaboratif permet, l'utilisation et l'assemblage de composants « plug and play » et « sur étagères » développés par différentes compagnies au sein d'un même environnement.

Le troisième type d'outil s'adresse plutôt au modélisateur de procédés qui préfère manipuler des équations et des variables ou un langage de modélisation. Les langages de modélisation permettent en effet à l'utilisateur de coder ses propres modèles à partir de quelques concepts de base (modèles, équations, variables, etc.). Ils sont extrêmement flexibles et plus généraux car non dédiés à un domaine particulier. Le langage est extensible, c'est-à-dire que l'utilisateur peut définir ses propres types par héritage ou agrégation. La plupart des langages de modélisation sont généraux, c'est-à-dire qu'ils permettent de représenter et simuler tous les systèmes physiques.

Enfin, un dernier type d'outils repose sur une approche orientée objet et s'adresse au développeur pour qui les concepts orientés objet lui sont familiers. Ce dernier manipule des objets élémentaires et génériques qui ne sont pas nécessairement des appareils réels, mais qui peuvent également être des entités abstraites et génériques. La modélisation qui en résulte est alors intuitive et très flexible. Les objets sont en effet plus fins que les modules et le développeur reste capable de manipuler les variables et les équations qui les représentent. La philosophie objet permet également de générer des entités facilement réutilisables pour des développements futurs et conduit à long terme à des temps et coûts de développement réduits. Enfin, soulignons que les objets déjà créés peuvent être manipulés sous forme d'une bibliothèque ou via une interface graphique, définissant alors un environnement de simulation.

C'est dans ce cadre que s'inscrit le développement de la plate-forme PrODHyS. Elle s'appuie sur la philosophie objet et fournit un ensemble de composants généraux, extensibles et réutilisables afin de construire des éléments plus spécifiques.

Pour plus d'information sur cet aspect, Perret (2003) propose une liste, non exhaustive des outils existants dans le domaine plus particulier des systèmes dynamiques hybrides en mettant notamment l'accent sur leurs capacités de modélisation et de simulation. La liste étant longue, elle s'est limitée aux outils les plus récents et les plus couramment rencontrés. Elle propose également une comparaison entre ces outils, comparaison basée sur deux critères : capacité de modélisation et capacité de simulation.

V Conclusion

Au final ce chapitre a proposé un tour d'horizon général sur les systèmes dynamiques hybrides. Il a, dans un premier temps, positionné ce domaine d'étude par rapport aux systèmes à dynamique continue et discrète. Il a ensuite introduit les aspects « modélisation » et a proposé, dans ce contexte, un ensemble de formalismes permettant de les représenter. Les formalismes proposés pour représenter les systèmes dynamiques hybrides sont nombreux. Ils se différencient par la technique de modélisation adoptée qui repose soit sur une approche visant à étendre un modèle discret ou un modèle continu, soit sur une approche visant à utiliser un modèle mixte. La technique de modélisation consistant à étendre un formalisme initialement prévu pour le domaine du discret ou du continu a montré ses limites. D'une part, elle privilégie l'une des deux dynamiques au dépend de l'autre ; or, les systèmes réels ont des comportements complexes qui nécessitent une modélisation précise tant au niveau de la partie continue que de la partie discrète. D'autre part, elle apporte peu de clarté et de lisibilité dans la délimitation entre la vision discrète et la vision continue du système. Par ailleurs, elle ne respecte pas certains principes d'évolution de marquage classiquement attribués aux réseaux de Petri : c'est le cas, par exemple, du réseau de Petri continu dans lequel deux transitions en conflit peuvent être franchies simultanément. Enfin, cette technique génère souvent des modèles à caractère spécifique et dont les applications sont réduites.

En revanche, la technique de modélisation reposant sur une approche mixte est plus générale et offre un potentiel de modélisation élevé dans chaque domaine (continu et discret). Par ailleurs, elle permet d'utiliser, pour la partie événementielle, des modèles de haut niveau.

Cet état de l'art sur les SDH, proposé dans ce chapitre, nous permet ainsi d'aborder plus aisément la problématique de cette étude, qui est d'optimiser les systèmes dynamiques hybrides. La finalité est d'ajouter la partie optimisation à la plateforme PRODHyS, développée dans notre laboratoire et présentée au chapitre suivant.

CHAPITRE 2

Méthodes d'Optimisation Dynamique

Nomenclature du Chapitre 2

- C : fonction globale de coût (fonction objectif)
 C_{eq} : contrainte équivalente
 C_j : vecteur des contraintes sur les variables d'état, de dimension n_c
 F : dynamique du couple état-état adjoint par le système Hamiltonien
 f : système des équations différentielles et algébriques
 G : vecteur des fonctions de tir
 g : fonction pour calculer la partie de Mayer de la fonction de coût
 g_1 : fonction pour la détection d'événements d'état
 h : fonction pour calculer la partie de Lagrange de la fonction de coût
 h_i : longueur du sous intervalle i dans la méthode simultanée
 H : hamiltonien
 k : fonction de pénalisation
 k_1 : point de collocation
 L : longueur du sous intervalle de discrétisation par la méthode PDI
 M : nombres de valeurs prises pour chaque fonction de contrôle
 N : nombre de fonctions de contrôle admissibles pour la méthode PDI
 P : nombre de subdivisions de l'intervalle de temps
 p : vecteur adjoint
 p' : vecteur des paramètres opératoires
 p^0 : réel
 q : ordre des polynômes
 r_j : région de contrôle « admissible » à l'itération j
 S : variable d'écart
 T : fonction de tir
 t : variable indépendante d'intégration (ici le temps)
 u : vecteur des fonctions de contrôle
 w : valeur initiale pour l'adjoint
 x : vecteur des variables d'état (inconnues) de dimension n
 \dot{x} : vecteur des dérivées premières des variables d'état
 x_1 : variables différentielles du vecteur des variables d'état
 x_2 : variables algébriques du vecteur des variables d'état
 x_{eq} : variable complémentaire pour la prise en compte des contraintes
 y : couple état-état adjoint
 \dot{y} : vecteur des dérivées premières du couple état-état adjoint

γ : conditions initiales ou finales

γ' : facteur de contraction

Ψ : fonction pour calculer les conditions de transversalité

β : valeur initiale pour y

λ : réel positif (très grand)

η application qui à y associe l'expression du contrôle optimal issue de la minimisation du Hamiltonien

ϑ : polynôme de Lagrange d'ordre q

χ : polynôme d'ordre q

Indices :

0 : condition initiale

f : condition finale

I Introduction

Les origines de l'optimisation des procédés de génie chimique, apparaissent au début de l'après guerre : Denbigh (1944), Bilous et Amundson (1956 a et b). Les études se sont ensuite enchaînées, bénéficiant des progrès de la modélisation, des méthodes mathématiques d'optimisation et surtout des techniques de calcul et de l'informatique. Il y a ainsi un parallèle évident entre les évolutions des études de simulation et celles d'optimisation, ces dernières intégrant des modèles de plus en plus complexes, initialement développés pour la simulation. Historiquement, les limitations de temps et de puissance de calcul ont restreint les travaux d'optimisation de procédés à l'étude d'une unité unique modélisée de façon simple. De nos jours, les travaux se focalisent sur des systèmes de plus en plus complexes

En général, pour les procédés de production la détermination des conditions optimales, afin d'en améliorer les performances, nécessite l'emploi de méthodes mathématiques complexes. L'introduction de méthodologies d'optimisation dynamique dans le domaine de l'industrie chimique s'est concentrée sur deux applications fondamentales : le contrôle optimal qui consiste en la détermination des conditions opératoires (variables de contrôle) optimales d'un procédé et le dimensionnement optimal permettant de choisir la configuration optimale du procédé de production selon l'objectif fixé. Dans certains cas, il est possible d'envisager le couplage design et contrôle optimal comme le démontrent les travaux de Montealegre (2005).

Remarque : L'optimisation dynamique touche une troisième application, certes plus limitée mais d'importance : l'identification paramétrique.

Le département PSI, intégré au sein du Laboratoire de Génie Chimique, développe depuis de nombreuses années des travaux de recherche relatifs à la modélisation et la simulation de procédés. L'un des résultats de ce travail est le développement progressif d'une plate-forme de simulation nommée PrODHyS (Process Object Dynamic Hybrid Simulator). Le but de sa création est de mettre en place un modèle hybride capable de traiter aussi bien les aspects discrets que continus d'un système industriel complexe. Cette plate-forme englobe différents modules développés depuis une quinzaine d'années au travers de plusieurs thèses. PrODHyS n'est pas une simple concaténation de travaux de recherche mais son développement est structuré grâce à l'utilisation d'outils logiciels de conception et de développement (UML...). La plate-forme PrODHyS permet une modélisation modulaire et hiérarchique des procédés. Une de ses originalités réside dans sa structure orientée-objet. En effet, elle est conçue comme une bibliothèque de classes dont l'objectif principal est de fournir des composants logiciels généraux et réutilisables. Ceux-ci peuvent alors être dérivés afin de construire des éléments spécifiques plus complexes en exploitant les mécanismes propres aux approches objets (composition, héritage, généricité, etc.).

PrODHyS est composée de trois modules principaux : un noyau numérique, une base de données thermodynamiques et une bibliothèque d'appareils.

I.1 La Plate-Forme PrODHyS

Cette partie est dédiée à une description succincte de la plate-forme, Perret (2003) en donne une description détaillée.

La version actuelle de PrODHyS comprend 7 modules comme le décrit schématiquement la figure 2.1. Elle met en évidence les connexions entre les différents modules et fait apparaître une structure comprenant deux couches logicielles : modélisation et simulation. Ce découpage «modélisation», «simulation» présente plusieurs intérêts :

- Créer des plate-formes dédiées à des domaines d'application divers en adaptant la couche «modélisation» adéquate.
- Permettre l'accès à différents types d'utilisateurs : développeur d'applications, un modélisateur...

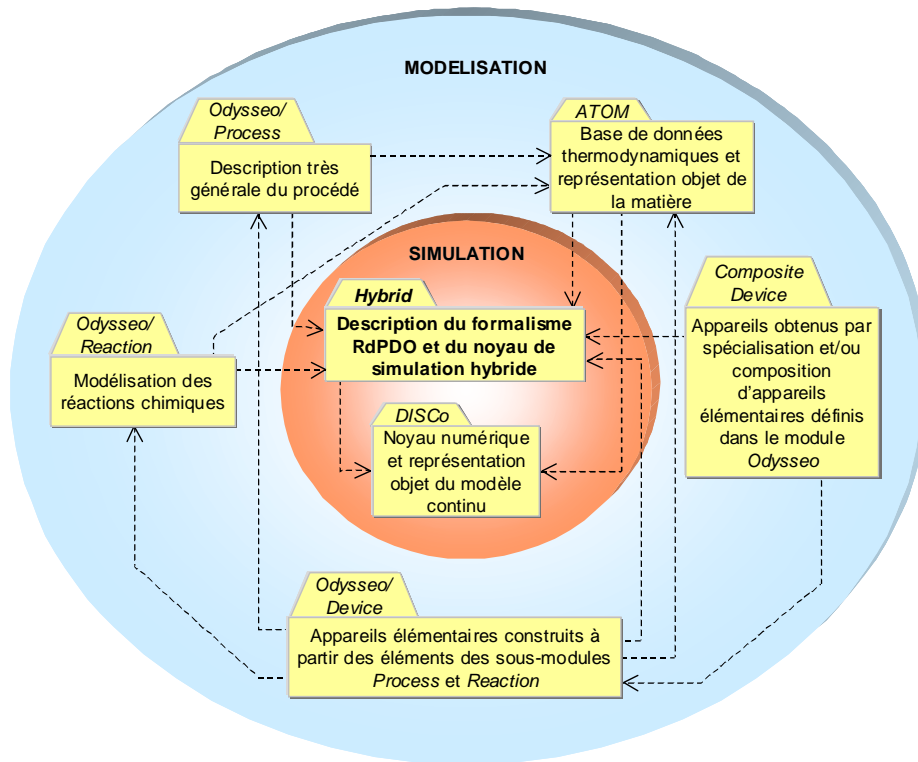


Figure 2.1 : Structure Logicielle de la plate-forme PrODHyS

La partie la plus interne de la plate-forme correspond au noyau de simulation. Elle fournit les éléments de base pour simuler n'importe quel système dynamique. Actuellement elle comprend :

- Le module **DISCo** qui constitue le noyau numérique, il fournit un ensemble de solveurs et d'intégrateurs EDA et EANL (Sargousse (1999), Le Lann (1999)).
- Le module **Hybrid** qui contient les classes nécessaires à la description du formalisme Réseaux de Pétri Différentiels à Objets ainsi que le noyau de simulation hybride (Perret (2003)).

La couche supérieure correspond à un ensemble de classes dédiées à la modélisation des procédés. Elle s'appuie sur la couche simulation et comprend des modules utiles lors de la construction d'un programme de simulation :

- Le module **ATOM** qui est la base de données thermodynamiques du système. Il s'appuie sur une représentation objet de la matière et permet le calcul de propriétés thermodynamiques (Jourda (1996), Jourda et al. (1996)).
- Le module **Odysseo** qui regroupe des entités élémentaires et génériques pour modéliser un procédé (Moysse (2000)). Il se compose de trois sous modules : *Process* (ensemble de classes pour une description générale du procédé), *Reaction* (modélisation de réactions chimiques), *Device* (appareils élémentaires du procédé).

- Le module **CompositeDevice** qui comprend des appareils obtenus par spécialisation et/ou composition d'appareils élémentaires définis dans **Odysseo**.

Un des premiers objectifs de travaux sur PrODHyS fut d'élaborer et de mettre en œuvre un formalisme suffisamment général pour permettre de représenter la plupart des systèmes dynamiques hybrides. Par conséquent, il doit pouvoir s'appliquer à différents secteurs d'activités comme l'automatique, la robotique, la mécanique, l'électronique, etc. Pour notre part, les exemples d'application concernent essentiellement le domaine du génie des procédés. Si l'on se reporte à la figure 2.1 représentant l'architecture générale de la plate-forme PrODHyS, on constate que les modèles discrets et continus et les interactions entre les deux sont gérés au niveau du noyau de simulation. Ce dernier doit donc être capable de simuler n'importe quel système sous réserve de lui transmettre un modèle en adéquation avec le formalisme proposé. Pour pouvoir utiliser ce modèle dans des domaines variés, il doit décrire de façon rigoureuse et précise aussi bien les composantes continues que discrètes des systèmes à simuler (Perret (2003)).

Une façon rigoureuse de représenter les aspects continus est d'utiliser des systèmes d'équations différentielles algébriques. Compte tenu de la complexité des phénomènes physico-chimiques rencontrés en génie des procédés, ils sont donc adoptés dans notre modèle pour simuler l'évolution continue des variables d'état. Notons que cette solution implique la présence d'un solveur d'équations différentielles algébriques possédant de nombreuses fonctionnalités (réduction d'index, recherche de conditions initiales cohérentes, détection d'événements, etc.), ce qui est effectivement le cas du module DISCo intégré à la plate-forme. La gestion de la partie continue est ainsi bien établie.

En revanche, de nombreux phénomènes discrets intrinsèques aux systèmes eux-mêmes (changements de phase par exemple) ou résultant du contrôle, de la commande ou de la supervision, ne peuvent être traités de façon simpliste en intégrant, par exemple, des variables booléennes au sein du modèle continu. Par conséquent, un modèle discret doit être mis en place afin de permettre une description aussi fidèle que possible de la composante événementielle du système. Un modèle discret basé sur les réseaux de Petri (RdP) semble bien approprié puisqu'il permet de représenter efficacement le séquençage, mais aussi les mécanismes de synchronisation, de parallélisme, d'affectation des ressources ou d'exclusion mutuelle.

La dynamique du système pouvant être représentée rigoureusement par l'utilisation conjointe d'un modèle continu basé sur les EDA et d'un modèle discret basé sur les RdP, il fallait donc ajouter un dernier concept pour décrire les aspects statiques et structurels du système. Pour combler ce manque, l'utilisation de l'approche objet est apparue tout à fait appropriée. Les

mécanismes de spécialisation, de composition et d'encapsulation qui lui sont propres engendrent en effet une description de haut niveau et permettent ainsi de réduire la taille du modèle et d'en améliorer la lisibilité.

Le formalisme RdPDO tire profit des RdP mixtes pour la description des aspects dynamiques et s'inspire du formalisme RdP à objets pour la description des aspects statiques. La puissance de description de la structure de donnée est ainsi augmentée puisque le jeton est défini par une classe d'objet encapsulant à la fois des données sous forme d'attributs et des méthodes qui traitent ces données.

L'intégration de l'ensemble des concepts décrits précédemment a conduit au développement du formalisme nommé Réseau de Petri Différentiel à Objets (Perret (2003)).

Ce travail de recherche est une brique supplémentaire dans l'évolution de la plate-forme PrODHyS, avec l'ajout d'un module d'optimisation dynamique et de contrôle optimal dans la couche simulation. Mais avant de présenter en détail ce nouveau module, il est nécessaire de décrire le noyau numérique avec lequel il a un lien direct.

1.2 Le Noyau Numérique

Le solveur

La modélisation réaliste et la simulation dynamique robuste des procédés conduisent souvent à des systèmes d'équations différentielles et algébriques (EDA). Les équations différentielles ordinaires (EDO) proviennent de l'application des lois de bilans en régime transitoire pour la conservation de la matière, de l'énergie et de la quantité de mouvement. Les équations algébriques non linéaires (EANL), quant à elles, découlent de relations constitutives ou phénoménologiques : thermodynamique, équilibres physico-chimiques, conditions technologiques....

Originellement, trois techniques ont été développées pour le traitement de tels systèmes :

- La reformulation et la résolution du système sous une forme purement algébrique par discrétisation des équations différentielles.
- La reformulation du système sous une forme purement différentielle par différentiation des équations algébriques. La résolution s'effectue alors au moyen d'un intégrateur classique d'équations différentielles.
- Le découplage des équations différentielles et algébriques. Dans un premier temps, les équations différentielles sont résolues pour un jeu de variables algébriques données, par l'intermédiaire d'un intégrateur d'équations différentielles. Les équations algébriques sont ensuite résolues de façon itérative : méthode de newton, quasi-newton, ...

Ces techniques de résolution, outre les lourdeurs intrinsèques qu'elles présentent, nécessitent l'utilisation d'outils robustes et fiables de discrétisation et d'intégration des équations différentielles. Ces inconvénients ont ainsi longtemps limité les travaux de simulation et d'optimisation dynamique à des représentations simplifiées. Gear (1971), en présentant un algorithme permettant d'intégrer le système sous sa forme originelle, a révolutionné le traitement des systèmes algébro-différentiels. Cet algorithme de type prédicteur-correcteur basé sur la différentiation rétrograde autorise en effet le traitement des systèmes algébro-différentiels sans manipulation d'aucune sorte sur les équations ni discrimination entre les variables. Gear a ainsi démontré qu'il est non seulement possible mais en général avantageux de conserver le système sous sa forme originelle.

De manière générale, la structure d'un modèle dynamique peut se représenter sous la forme implicite suivante :

$$\begin{aligned} f(\dot{x}, x, p, u, t) &= 0 \\ x(t_0=0) &= x_0 \end{aligned} \quad (2.0)$$

Avec f : jeu d'équations EDA

x : inconnues du système de dimension n

\dot{x} : vecteur des dérivées premières par rapport à la variable indépendante (t)

p : ensemble de paramètres opératoires du système

u : vecteur des paramètres de contrôle

t : variable indépendante (généralement le temps)

Une fois l'étape de l'écriture du modèle mathématique passée, vient l'étape de résolution qui s'appuie sur une stratégie de résolution globale. Dans le module DISCO trois solveurs ont été implantés :

- 1- Un solveur d'équations algébriques linéaires choisis dans une bibliothèque commerciale de méthodes numériques.
- 2- Un solveur d'équations algébriques non linéaires.
- 3- Un solveur de systèmes algébro-différentiels. Il a été conçu dans le but de résoudre des systèmes creux de grande taille (classe de problèmes importante en génie des procédés). Le solveur est basé sur la méthode de Gear (Gear (1971)). Ses principales qualités sont sa grande robustesse, sa capacité à passer des discontinuités, son aptitude à traiter des systèmes de grande taille, un algorithme de réduction d'index et un calcul automatique de conditions initiales cohérentes. Une autre de ses particularités est de posséder un système de détection des événements.

Le type de résolution effectué par DISCo est une résolution globale : toutes les équations des différents composants sont regroupées puis résolues globalement à chaque pas de temps. Bien que plus lourde à mettre en œuvre que la simulation séquentielle (pour laquelle les équations des composants sont résolues séparément), cette approche est plus rigoureuse.

L'intégrateur numérique utilisé repose sur trois formes de systèmes d'équations : équations différentielles ordinaires (EDO), équations différentielles algébriques (EDA), équations aux dérivées partielles (EDP). Le solveur continu est capable de résoudre des EDO directement. Des EDP peuvent également être résolues par DISCo : il convient de discrétiser un tel système et fusionner les modèles. Quant aux EDA, DISCo peut résoudre de tels systèmes d'index inférieur ou égal à un.

La précision obtenue dans la résolution des équations du système continu est assujettie à la sévérité préconisée par l'utilisateur en termes de tolérances. Elle ne peut être limitée que par la précision de l'ordinateur.

L'intégrateur DISCo est à ordre et à pas variables. Le changement d'ordre et/ou de pas est conditionné par le calcul de l'erreur de troncature.

Remarque : A l'heure actuelle, le module DISCo ne traite pas les équations différentielles partielles

La réduction d'index

L'index de différentiation d'un modèle est le nombre minimum de fois que tout ou partie du système doit être dérivé en fonction de la variable indépendante de façon à exprimer \dot{x} comme une fonction continue de x et t . Les systèmes d'index 0 et 1 ne posent généralement pas de problème au solveur. En revanche, lorsque cet index devient supérieur à 1, l'intégration est plus compliquée. Pour pallier cette difficulté, il existe des algorithmes qui permettent de revenir à un problème d'index 1 (Pantelides (1988), Mattsson et al (2000)). Cependant en génie des procédés une bonne modélisation conduit dans la plupart des cas à des systèmes d'index rarement supérieur à 2.

Détection d'événements

Les événements temporels sont générés à des instants prédéfinis, la détection de leur occurrence ne pose pas de problèmes. Par contre, les événements d'états surviennent lorsque les variables d'états franchissent certains seuils, donc l'instant de leur occurrence n'est pas connu. Il faut dans un premier temps détecter l'événement puis le temps précis d'occurrence doit être localisé. Un événement d'état est généralement décrit par une fonction qui dépend des variables d'états continues et dont le changement de signe traduit la présence de l'événement. Soit $g_1(x,t)$ une telle fonction. La gestion de l'événement est effectuée

conjointement à l'intégration numérique afin de vérifier à chaque pas d'intégration si un événement a eu lieu. La procédure est la suivante :

- Le solveur évalue les fonction $g_1(x_n, t_n)$ à chaque pas d'intégration t_n .
- Si le signe de $g_1(x_n, t_n)$ est différent de celui de $g_1(x_{n-1}, t_{n-1})$ alors un (ou plusieurs) événement a eu lieu dans l'intervalle de temps $[t_{n-1}, t_n]$. Dans ce cas un algorithme de résolution de la (ou des) racine de la fonction $g_1(x, t)=0$ est mis en œuvre sur cet intervalle. Il permet de déterminer avec exactitude l'occurrence de l'événement (ou du premier). L'algorithme utilise généralement une interpolation de x dans l'intervalle afin de calculer sa valeur exacte en tous les points requis, pour gagner en temps calcul.
- L'intégration s'arrête au temps d'occurrence de l'événement.

I.3 Conclusion

Conçue selon une approche objet, *PrODHyS* fournit aussi une bibliothèque de composants logiciels autonomes qu'il suffit d'assembler ou de spécialiser pour développer un appareil spécifique. Pour un développeur, l'exploitation de ces composants élémentaires permet d'accélérer le processus de conception et d'implémentation d'un nouvel appareil. Pour un utilisateur, l'exploitation comme des "boîtes noires", des appareils prédéfinis offerts facilite la mise en place de campagnes de simulation. Ces outils font de *PrODHyS* un outil opérationnel et évolutif.

Cette plate-forme possède plusieurs points forts dont les plus importants sont :

- Elle permet tout d'abord une représentation rigoureuse et explicite des composantes continues et discrètes ; en effet, les réseaux de Petri engendrent une description efficace des contraintes de synchronisation, de parallélisme et de séquençement et les équations différentielles algébriques conduisent à une représentation précise et fiable de la dynamique continue (à condition de disposer d'un solveur suffisamment puissant notamment pour la recherche et la gestion des événements ainsi que pour le calcul des conditions initiales).
- L'« approche combinée étendue » permet une intégration sur plusieurs niveaux des réseaux de Petri et des objets. Ceci permet de réduire la taille du modèle, de découpler la complexité du système, d'organiser l'information, d'améliorer la lisibilité du modèle et par conséquent d'en faciliter sa réutilisation.
- De manière générale, l'utilisation des concepts objet engendre une description de haut niveau, favorise la modularité des modèles, permet la création d'entités génériques, extensibles et réutilisables (en exploitant notamment les mécanismes de spécialisation et de composition).

Après cette description, même succincte, de la plate-forme *PrODHyS*, il apparaît clairement la nécessité de la faire évoluer pour en assurer la pérennité. Parmi les multiples voies possibles,

il est intéressant de regarder la couche simulation. L'évolution de cette couche peut être envisagée dans plusieurs directions, allant de l'amélioration des caractéristiques actuelles à l'ajout de nouvelles fonctionnalités : intégration d'un module d'optimisation dynamique et de contrôle optimal. Ce module sera évidemment en étroite liaison avec le module DISCo.

II Le problème de Contrôle Optimal

II.1 Définition

La première étape du problème du contrôle optimal consiste à déterminer un ensemble de trajectoire joignant un état initial (souvent connu) à une cible finale (connue ou non). Ensuite il faut chercher parmi cet ensemble de trajectoires celle(s) qui le réalise en un coût minimal. La première étape revient à savoir si la cible est atteignable : problème de contrôlabilité. Dans la suite de ce mémoire, l'ensemble des trajectoires admissibles est, par hypothèse, non vide. Pour des considérations mathématiques plus pointues sur le sujet se conférer à Trélat (2005).

Le problème général s'écrit sous la forme :

$$\begin{aligned} \dot{x} &= f(x, p', u, t) \\ \gamma_0(t_0, x(t_0)) &= 0 \text{ (Conditions initiales)} \\ \text{Et éventuellement des conditions finales } \gamma_f(t_f, x(t_f)) &= 0 \end{aligned} \tag{2.1}$$

Où f est une fonction de classe C^1 sur l'intervalle considéré et u appartient à un sous ensemble U (non vide). Le système (2.1) peut être éventuellement soumis à des contraintes sur l'état. En génie des procédés ces contraintes sont fréquentes, elles traduisent par exemple des contraintes de sécurité, de fonctionnement : certaines variables d'état doivent rester dans un intervalle fixé (gamme de températures, de débits...). Ces contraintes peuvent se mettre sous la forme :

$$C_j(x, p, u) \leq 0 \quad j = 1, \dots, n_c \tag{2.2}$$

Soit une fonction h de classe C^1 sur l'intervalle considéré, et g une fonction continue. Pour tout contrôle admissible u , le coût de sa trajectoire x_u associée sur l'intervalle de temps $[t_0, t_f]$ est :

$$C(t_f, u) = \int_{t_0}^{t_f} h(t, x_u(t), p', u(t)) dt + g(t_f, x_u(t_f), p', u(t_f)) \tag{2.3}$$

Le but est de trouver la trajectoire x_u qui minimise cette fonction coût : si une telle trajectoire existe, elle est appelée trajectoire optimale pour ce coût. Pour le coût $C(t_f, u)$, le temps final t_f peut être fixé : *problème de contrôle optimal à temps final fixé* sinon le problème est dit à *temps final non fixé*.

Remarque : Pour la fonction coût du type (2.3), le problème est qualifié du type Mayer-Lagrange. Une fonction coût avec seulement la partie intégrale est appelée problème de Lagrange et problème de Mayer dans le cas complémentaire. Toutefois, tout problème peut se ramener à un problème du type Mayer. En effet, il suffit d'ajouter une $n+1^{\text{ème}}$ variable et l'équation $\dot{x}_{n+1} = h(t, x(t), p', u(t))$ au système initial (2.1).

II.2 Principe du Maximum de Pontryagin

Lors de la description d'un problème de contrôle optimal il est incontournable de présenter le principe du Maximum de Pontryagin (1974). Il existe plusieurs versions de ce principe (dégradées de la version générale) en fonction de la configuration du problème. Suivant le type de la fonction coût, si le temps final est fixé ou non, l'éventuelle présence de contrainte sur le contrôle et/ou l'état, l'écriture des équations varie (Trélat (2005)). Lorsque le problème initial n'est soumis à aucune contrainte (contrôle et état), le principe devient le principe du maximum faible. Lorsque les fonctions de contrôles sont contraintes l'énoncé est le suivant :

Enoncé du principe du maximum

Soit le système de contrôle (2.1) avec des conditions initiales $\gamma_0(0, x(0)) = 0$ et finales $\gamma_f(t_f, x(t_f)) = 0$, ainsi que le coût associé du contrôle u (2.3). Le problème de contrôle optimal consiste à déterminer la trajectoire joignant le point initial au point final (fixe ou libre) tout en minimisant la fonction de coût.

Si u (appartenant à l'ensemble des contrôles admissibles) est optimal sur l'intervalle de temps, alors il existe une application p absolument continue, appelée vecteur adjoint, et un réel $p^0 \leq 0$ tels que le couple (p, p^0) est non trivial et que $\forall t$:

$$\begin{aligned} \dot{x} &= \frac{\partial H}{\partial p}(t, x(t), p(t), p^0, p', u(t)) \\ \dot{p} &= -\frac{\partial H}{\partial x}(t, x(t), p(t), p^0, p', u(t)) \end{aligned} \quad (2.4)$$

où H est le Hamiltonien du système :

$$H(t, x(t), p(t), p^0, p', u(t)) = p^0 * h(t, x(t), p', u(t)) + (p(t) \bullet f(t, x(t), p', u(t)))$$

Avec la condition de maximisation : u solution de $\text{Max}_{v \in U} H(t, x(t), p(t), p^0, p', v)$ (2.5)

Le vecteur adjoint doit être construit de manière à vérifier les conditions de transversalité aux deux extrémités :

$$\gamma_0(0, x(0)) = 0 \quad \text{Conditions Initiales} \quad (2.6a)$$

$$p(0) = -\frac{\partial \Psi}{\partial x_0}(0, x(0), t^f, x(t_f), \mu_0, \mu_1) \quad (2.7a)$$

$$\gamma_1(t_f, x(t_f)) = 0 \quad \text{Conditions Finales} \quad (2.6b)$$

$$p(t^f) = \frac{\partial \Psi}{\partial x_f}(0, x(0), t^f, x(t_f), \mu_0, \mu_1) \quad (2.7b)$$

Avec :

$$\Psi(0, x(0), t^f, x(t_f), \mu_0, \mu_1) = g(t_f, x(t_f), p', u(t_f)) + [\gamma_0(0, x(0)) \bullet \mu_0] + [\gamma_1(t_f, x(t_f)) \bullet \mu_1]$$

Les conditions (2.6b) et (2.7b) sont nommées conditions de transversalité sur le vecteur adjoint.

Si de plus le temps final n'est pas fixé il y a une condition supplémentaire :

$$\text{Max}_{v \in U} H(t, x(t), p(t), p^0, p', v) = -p^0 \frac{\partial g}{\partial t}(t_f, x(t_f), p', u(t_f)) \quad (2.8)$$

Cette dernière est appelée condition de transversalité sur l'adjoint.

Remarques :

-1- La convention $p^0 \leq 0$ conduit au principe du maximum. La convention $p^0 \geq 0$ conduit au principe du minimum, i.e. la condition (2.5) devient une condition de minimum.

-2- Le cas dit normal revient à diviser tout le système adjoint par p^0 .

-3- Lorsqu'il n'y a pas de contrainte sur le contrôle, la condition de maximum (2.5)

devient $\frac{\partial H}{\partial u}(t, x(t), p(t), p^0, p', u(t)) = 0$, ce qui redonne le principe du maximum faible.

Le principe du maximum de Pontryagin énoncé précédemment, ne prend pas en compte d'éventuelles contraintes sur l'état. Dans ce cas, la démonstration du principe ainsi que son énoncé deviennent très difficiles. Il existe plusieurs versions du principe du maximum avec contraintes d'état : Hartl et al (1995), Miele (1996). La théorie d'un tel problème n'est pas abordée dans ce manuscrit. Par rapport au principe du maximum classique (énoncé ci-dessus), la difficulté réside dans le fait que la présence des contraintes sur l'état peut rendre le vecteur adjoint discontinu. Alors il est nécessaire de rajouter des conditions de saut ou de jonction.

Maintenant que le problème général est posé, il va falloir s'attacher à le résoudre. Il existe deux stratégies principales de résolution : les méthodes indirectes obtenues par application du principe du maximum et les méthodes directes qui impliquent traditionnellement une discrétisation totale (contrôle et variables d'état) ou partielle (contrôle uniquement) du

problème. Les principes de ces deux grandes voies de résolution sont présentés dans les paragraphes suivants.

III Les méthodes indirectes

L'application du principe du maximum conduit à un problème aux valeurs limites qui, le plus souvent, est résolu numériquement par des méthodes de tir simple ou tir multiple. La présentation des méthodes de tir est faite sans prendre en compte les contraintes sur l'état. Le système étudié résulte de la concaténation des équations (2.1) avec la fonction de coût (2.3).

III.1 Méthode de tir simple

Le principe général de cette méthode consiste à transformer le problème de contrôle optimal en la recherche de zéro de la fonction de tir (non linéaire) associée. Il n'y a pas de discrétisation explicite même si la méthode requiert tout de même l'intégration du système EDA.

Dans un premier temps, le temps final est supposé fixé et connu pour simplifier la présentation. L'application du principe du minimum fournit une condition nécessaire d'optimalité. Si grâce à la condition de minimum (2.5), il est possible d'exprimer le contrôle extrême en fonction de $(x(t), p(t))$ alors le système initial équivaut à résoudre le problème aux valeurs limites suivants (problèmes aux deux bouts) :

$$\begin{aligned} \dot{y} &= F(y, \eta(y)) && \text{pour } t \in [t_0, t_f], \text{ avec } y=(x, p) \text{ couple état-état adjoint.} \\ &F \text{ dynamique du couple état - état adjoint par le système Hamiltonien.} \\ &\eta \text{ application qui à } y \text{ associe l'expression du contrôle optimal issue de la} \\ &\text{minimisation du Hamiltonien.} \\ C_0(y(0)) &= 0 && \text{Conditions aux limites en } 0 \\ C_f(y(t_f)) &= 0 && \text{Conditions aux limites en } t_f \end{aligned} \tag{2.9}$$

Remarque : Les conditions C_0 et C_f représentent les conditions de transversalité de l'énoncé du minimum, à savoir : les conditions initiales et finales du système et des conditions sur l'état adjoint : (2.6) et (2.7).

Il est alors possible d'intégrer $y = (x, p)$, si la valeur initiale $y(t_0)$ est fixée, le problème à valeur initiale suivant est obtenu :

$$\begin{cases} \dot{y} = F(y) \\ y(t_0) = \beta \end{cases} \tag{2.10}$$

L'application appelée fonction de tir est alors introduite : à la valeur initiale β elle associe la valeur des conditions limites en t_f pour la solution correspondante à (2.10). En pratique les

condition initiales γ_0 du problème initial fixent une partie de $y(t_0)$, par conséquent l'inconnue de la fonction de tir est donc réduite à la partie manquante, noté w . Fréquemment, les condition initiales déterminent l'état initial $x(t_0)$, alors w se résume à la valeur initiale de l'état adjoint $p(t_0)$. La valeur de la fonction de tir est alors donnée par les conditions en t_f pour la solution y du problème (2.10) avec $y(t_0)=(x_0, w)$

$$T(w) = C_f(y'(t_f)) \tag{2.11}$$

Trouver une racine de T revient à résoudre le problème aux deux bouts (2.9) et donne une solution au problème initial. La méthode de tir se résume à résoudre l'équation $T(w)=0$.

Remarque : Les contraintes sur les variables d'état ne sont pas prises en compte dans cette présentation de la méthode, il faudrait revoir la formulation du principe. Car comme il a été précisé plus en amont elles génèrent des difficultés supplémentaires avec les méthodes indirectes.

Si l'hypothèse du temps final fixé est levée, t_f devient libre, il est possible de revenir à la situation précédente en considérant t_f comme une inconnue. Dans ce cas la dimension du système initial est augmentée en ajoutant l'équation supplémentaire $\frac{dt_f}{dt}=0$. Toutefois, dans ce cas il peut s'avérer préférable de considérer la condition de transversalité sur le Hamiltonien, comme le précise Trélat (2005).

III.2 Méthode de tir multiple

Le principe générale de la méthode est le même que celui du cas précédent : *transformation du problème de contrôle optimal en un problème de recherche de racines d'une fonction*. Mais contrairement au cas précédent, cette méthode découpe l'intervalle de temps $[t_0, t_f]$ en N sous intervalles $[t_i, t_{i+1}]$, ainsi les $y(t_i)$, au début de chaque sous intervalle, deviennent des inconnues (rappel $y=(x,p)$, avec p vecteur adjoint). Toujours dans un souci de simplicité de présentation de la méthode, le temps final est supposé connu. Soit un découpage de l'intervalle de temps : $t_0 < \dots < t_i < t_{i+1} < \dots < t_f$. La fonction y étant continue sur l'intervalle de temps, et de part le découpage il faut assurer la continuité de la fonction au point de subdivision :

$$y(t_i^-) = y(t_i^+) \tag{2.12}$$

Finalement, cela revient au problème :

$$\begin{aligned} \dot{y} &= F(t, y(t)) \\ \forall i \in \{2, \dots, N-1\} \quad C_i^l(t_i, y(t_i^-), y(t_i^+)) &= 0 \\ C_m^l(t_m, y(t_1), y(t_m)) &= 0 \end{aligned} \tag{2.13}$$

Avec $t_1 = t_0$ et $t_m = t_f$, et les C_j^1 sont les conditions de continuité et aux limites. De plus, si $y_i^+ = y(t_i^+)$ et $y(t_{i-1}, y_{i-1}^+)$ solution de :

$$\dot{y} = F(t, y(t)) \text{ avec comme condition initiale } y(t_{i-1}) = y_{i-1}^+$$

$$\text{De ce fait } y(t_i^-) = y(t_{i-1}, y_{i-1}^+).$$

Les conditions de continuité et aux limites s'écrivent :

$$\begin{aligned} \forall i \in \{2, \dots, N-1\} \quad C_i^1(t_i, y(t_{i-1}, y_{i-1}^+), y_i^+) &= 0 \\ C_m^1(t_m, y_1^+, y(t_m, t_{m-1}, y_{m-1}^+)) &= 0 \end{aligned} \quad (2.14)$$

Ces conditions sont vérifiées si :

$$G(Y) = \begin{pmatrix} C_N^1 \\ C_2^1 \\ \cdot \\ \cdot \\ C_{N-1}^1 \end{pmatrix} = 0 \quad (2.15)$$

Au final, la méthode se réduit à la recherche d'une racine d'une fonction G dont la dimension est égale au nombre de point de subdivision. Tout comme au paragraphe précédent, la recherche des racines pourra se réaliser à l'aide d'une méthode itérative de type Newton.

La conséquence directe de la subdivision est que le nombre d'inconnues est beaucoup plus important que dans le cas de la méthode de tir simple, mais elle offre la possibilité de paralléliser l'intégration du système initial. Un autre avantage de la subdivision est l'augmentation de la stabilité par rapport à la méthode précédente, stabilité d'autant plus accrue que le nombre de subdivision est important. En effet, dans la méthode de tir simple, les erreurs par rapport à la condition initiale évoluent exponentiellement en fonction de la taille de l'intervalle, Stoer et Bulirsch (1980).

Remarque : Avec cette méthode il est facile d'intégrer des temps de commutation avec éventuellement changement du système d'équations. Dans sa résolution le problème reste analogue au cas sans commutation mais la taille du système à résoudre (recherche de racines) est augmentée (Trélat (2005) pour plus de détails).

III.3 Autres Méthodes

Des méthodes basées sur le principe du maximum, celles de tir sont les plus utilisées toutefois, comme le précise Srinivasan et al (2003), il en existe d'autres. Pour une première catégorie de méthode, le problème est abordé par une paramétrisation des équations d'état et du système adjoint (ensemble du vecteur y). En général, ces variables sont supposées s'écrire

sous forme polynomiale. A partir de cette hypothèse deux stratégies différentes sont possibles :

- 1- *Discrétisation* : l'ensemble des équations du système EDA (2.9) sont discrétisées pour un nombre fini de points de temps (via collocation orthogonale). Au final, le système EDA (2.9) est transformé en un système d'équations algébriques non linéaires. Après initialisation, le système transformé est résolu avec une méthode du type Newton Raphson.
- 2- *Quasi-Linéarisation* : la deuxième stratégie consiste à linéariser le système (2.9) autour de la trajectoire courante. Au final, le système devient un système EDA linéaire. Le système linéaire est ensuite résolu, la solution devient alors la nouvelle trajectoire autour de laquelle le système est linéarisé à nouveau.

III.4 Conclusion

Les présentations des méthodes indirectes se basent sur l'hypothèse de non prise en compte des contraintes sur l'état. Or, c'est une hypothèse très pénalisante pour le domaine d'application visé, i.e. le génie des procédés, car dans la majorité des cas les variables d'état sont soumises à ce type de contraintes (sécurité, fonctionnement...). Lors de la résolution d'un problème de contrôle optimal avec contraintes sur l'état, la difficulté engendrée par la prise en compte de ces contraintes ne permet pas d'employer les méthodes indirectes comme le souligne notamment Biegler et Grossmann (2004). Comme il a été mentionné au paragraphe II.2, cette difficulté émerge à cause de problème de discontinuité (saut) sur le vecteur adjoint. Toutefois, dans certains cas, il est envisageable de résoudre avec succès un problème de contrôle optimal contraint : Bonnard et al (2003), Hartl et al (1995).

Comme le précise Trélat (2005), la méthode de pénalisation est un moyen simple pour prendre en compte les contraintes d'état dans le cas d'une méthode indirecte. Le principe de cette méthode, consiste en une manipulation de ces contraintes par une pondération, dans le but de les forcer à être vérifiées. Les contraintes imposent à l'état d'appartenir à un sous ensemble de \mathfrak{X}_n . Soit une fonction k sur \mathfrak{X}_n , nulle sur le sous ensemble admissible et strictement positive sur le complémentaire (la difficulté est de construire cette fonction). En

ajoutant à la fonction de coût (2.3), le scalaire $\lambda \int_0^{tf} k(x(t)) dt$, où $\lambda > 0$ est une pondération choisie

très grande, de façon à forcer la trajectoire à rester dans le sous ensemble défini par les contraintes. En effet, si $x(t)$ sort de ce sous ensemble admissible, et comme λ est très grand alors le coût correspondant sera conséquent et vraisemblablement la trajectoire ne sera pas optimale. Cette méthode est peu mise en œuvre à cause de la difficulté à construire k et sur l'hypothétique non optimalité de la trajectoire.

Comme présenté dans les paragraphes précédents, la mise en œuvre de méthodes indirectes nécessite la recherche de racines d'une fonction qui est classiquement réalisée par un algorithme de type Newton. Or, ce type de méthode (Newton) est particulièrement sensible aux conditions initiales. Il faut alors être capable de deviner les bonnes conditions initiales pour l'état et l'adjoint car dans certains cas le rayon de convergence est parfois très faible. Une possibilité pour pallier cet inconvénient est d'utiliser une méthode d'homotopie (ou de continuation). L'idée est de résoudre le problème difficile en partant d'un problème dégradé plus simple. Pour cela il faut introduire une fonction qui connecte ces deux problèmes. La première étape est de construire cette fonction, pour cela on introduit un paramètre homotopique tel que : quand il vaut 1 on a le problème initial et quand il vaut 0 le problème dégradé. Le but est de partir de la solution du problème facile et d'atteindre le problème difficile, en faisant varier graduellement le paramètre homotopique. Chaque problème intermédiaire utilise la solution précédente comme condition initiale. Pour faciliter l'incrémentement du paramètre homotopique, il est préférable que se soit un paramètre « naturel » du problème.

Remarques : -1-La méthode d'homotopie présentée est une méthode de continuation discrète, évolution discrète du paramètre. Toutefois il existe des méthodes de suivi de chemin complet : continuation différentielle et méthode simpliciale : Allgower et Georg (1990, 2000), Martinon (2005).
-2- Dans certains cas, une méthode directe (paragraphe suivant) peut être mise en place pour que le résultat puisse initialiser une méthode indirecte.

En contre partie, les méthodes indirectes sont d'une grande précision par rapport aux autres ce qui peut s'avérer un critère de choix important pour certains problèmes comme en aéronautique par exemple. Un autre avantage par rapport aux méthodes directes, est qu'elles nécessitent moins de mémoire et par conséquent deviennent efficaces si la dimension du problème EDA est trop grande. De plus, la méthode de tir multiple est parallélisable, de par le découpage de l'intervalle de temps.

En génie des procédés, le Principe du maximum de Pontryagin a été employé par Siebenthal et Aris (1964), Jackson et Senior (1968), Jackson et Obando (1971), dans le cas de réacteurs batch et Coward (1967a et b), Robinson (1969), Mayur et Jackson (1971), Kerkhof et Vissers (1978) pour la distillation discontinue. Dans le domaine du génie des procédés, l'utilisation de méthodes basées sur le principe du maximum est relativement ancienne car elles constituaient un frein technologique à l'évolution vers des travaux plus complexes. En effet, ces méthodes apparaissent peu adaptées à la résolution de problèmes d'optimisation dynamique faisant intervenir une grande variété de variables et de contraintes différentes.

IV Méthodes directes

Les études d'optimisation de procédés (basées sur des méthodes d'optimisation variationnelles ou le principe du maximum de Pontryagin) nécessitaient généralement un temps de calcul et des capacités de stockage numérique importants. Leur utilisation restreignait ainsi les problèmes d'optimisation à la résolution de modèles simplifiés. A partir du milieu des années 80, l'abandon de ces méthodes au profit de techniques de programmation non-linéaire (PNL) a permis de débloquer cette situation. Les méthodes de programmation non-linéaire présentent en effet la capacité de traiter des systèmes d'équations de grande dimension et ainsi d'utiliser des modèles particulièrement détaillés. Ce n'est toutefois qu'avec le développement de méthodes efficaces de programmation non-linéaire, et surtout de la méthode de programmation quadratique successive que l'optimisation dynamique des procédés a pu s'étendre vers des modèles complexes.

L'évolution des études de contrôle optimal de procédés discontinus s'est principalement orientée vers une augmentation de la complexité des modèles (description plus rigoureuse et détaillée) et des problèmes d'optimisation (traitement simultané de nombreuses variables et introduction de contraintes). Les premières études de contrôle optimal intégrant des modèles de représentation complexes et résolues à partir de techniques d'optimisation efficaces, apparaissent vers la fin des années 80 dans la littérature. Généralement ces études sont basées sur une modélisation EDA et permettent l'optimisation dynamique des procédés par l'intermédiaire de méthodes indirectes.

Le principal obstacle à la résolution du problème d'optimisation formé par les équations (2.1), (2.2) et (2.3) réside dans sa nature de dimension infinie. En effet, les variables d'état du système et les variables de contrôle, fonctions dépendantes du temps, font de ce problème, un problème d'optimisation dynamique de dimension infinie. L'utilisation d'une approche de programmation non-linéaire nécessite ainsi une étape préalable de transformation du problème général de dimension infinie en un problème de programmation non-linéaire de dimension finie. Dans ce contexte, il est usuel d'utiliser une technique de paramétrisation afin de réduire la dimension du problème. C'est ce que mettent en œuvre les méthodes indirectes.

Ces méthodes transforment le problème de contrôle optimal en un problème d'optimisation non linéaire en dimension finie. L'idée est d'appliquer des niveaux de discrétisation qui convertissent le problème originel continu dans le temps en un problème discret. La classification de ces méthodes en fonction du niveau de discrétisation, présentée par Biegler et Grossmann (2004), est reprise dans cette partie. Le premier niveau de discrétisation englobe les méthodes qui discrétisent seulement le profil de contrôle : discrétisation partielle. La seconde catégorie de méthodes discrétise à la fois le profil de contrôle et l'état : discrétisation totale. Quelle que soit la méthode directe, ses avantages principaux par rapport aux méthodes

indirectes sont sa facilité de mise en œuvre et d'introduction d'éventuelles contraintes sur l'état.

Dans le cadre d'une discrétisation partielle, le problème d'optimisation résultant peut être résolu par une méthode de programmation non linéaire (PNL) ou de programmation dynamique. Avec une méthode de programmation non linéaire, le système EDA est intégré à chaque itération du solveur PNL avec les nouvelles valeurs du vecteur de contrôle. Par contre le principal avantage de cette approche est qu'elle génère un système de petite taille pour le solveur PNL, comparativement à la méthode de discrétisation totale.

Les méthodes de discrétisation totale utilisent également une technique de PNL pour résoudre le système discret mais contrairement aux méthodes de discrétisation partielle, le système EDA est intégré seulement une fois, à l'optimum. De plus, elles ont une meilleure stabilité que les méthodes de discrétisation partielle, surtout en présence de mode dynamique instable. En contre partie, la taille du PNL à résoudre est fortement accrue suite à la discrétisation totale.

IV.1 Approche séquentielle

Pour les approches séquentielles, une discrétisation partielle est mise en œuvre, touchant uniquement les fonctions de contrôle. Ces fonctions sont dans la majorité des cas représentées par des polynômes. Dans cette approche, les conditions initiales et un ensemble de paramètres de contrôle étant donnés, le système EDA est intégré de façon classique à chaque itération de la méthode d'optimisation : méthode de programmation non linéaire. La fonction objectif est évaluée, et elle est utilisée par le solveur de programmation non linéaire pour trouver les paramètres de contrôle optimaux. La caractéristique principale de cette méthode provient du fait qu'une solution de l'EDA est obtenue à chaque itération de l'optimisation. La satisfaction des équations différentielles à chaque itération conduit à qualifier ces approches de « chemin réalisable ». L'algorithme simplifié de ces approches se résume comme suit :

- 1- Paramétrisation des entrées par un nombre fini de variables de décision
- 2- Choix de valeurs initiales de ces variables
- 3- Intégration du système jusqu'au temps final et calcul de la fonction objectif et des éventuelles contraintes.
- 4- Utilisation d'un algorithme d'optimisation pour recalculer les variables de décision. Les deux dernières étapes sont répétées jusqu'à minimisation de la fonction objectif.

Lors de la résolution du problème de programmation non linéaire, les gradients de la fonction objectif et des contraintes par rapport aux variables de décision (paramètres de contrôle) se

calculent généralement à l'aide des équations de sensibilités. Cet aspect sera détaillé dans le chapitre suivant au paragraphe III.1.

Dans cette approche l'horizon de temps est subdivisé en sous intervalles de contrôle. Sur chaque sous intervalle, les variables de contrôle sont représentées par des constantes, des droites, des polynômes (la qualité de la solution est directement corrélée à la paramétrisation du contrôle Logdson et Biegler (1989)).

Remarque : Si la fonction de contrôle est approximée par des constantes sur des sous intervalles de temps identiques, alors la méthode est nommée : Paramétrisation des vecteurs de contrôle (Control Vector Parameterization).

Remarque : Plusieurs schémas différents de paramétrisation apparaissent dans la littérature. Bien que la plupart des problèmes soit résolue par une paramétrisation à base de polynômes de Lagrange (Vassiliadis et al, (1994a) et (1994b)), cependant quelques travaux adoptent une paramétrisation plus complexe, à partir de fonctions exponentielles (Farhat et al, (1990)).

En dépit d'une implémentation directe de cette méthode, elle est lente surtout lors de la prise en compte de contraintes inégalités (Bell et al (1996)). Ceci est principalement dû au fait que c'est une méthode à chemin réalisable, par conséquent elle nécessite à chaque itération l'intégration des équations différentielles qui est très coûteuse en temps calcul. Une fois de plus, toute la difficulté apparaît lors de la prise en compte des contraintes égalités ou inégalités. Dans ces méthodes, la majorité des techniques intègrent ces contraintes en mesurant la violation des contraintes sur l'horizon de temps puis en ajoutant une fonction de pénalisation (dépendant de cette mesure) à la fonction objectif ou en forçant ces contraintes directement à zéro grâce à une contrainte sur le temps final (Vassiliadis et al (1994a), (1994b)). Le principe de la fonction de pénalisation est exposé dans le paragraphe précédent. Pour éviter les difficultés numériques causées par la construction de la fonction de pénalisation, une solution est de convertir ces contraintes en une contrainte équivalente au point final.

Par exemple soit :

$$C_{eq}(x,u,p,t)=0 \quad (2.16)$$

Dans ce contexte une nouvelle variable différentielle est introduite :

$$x_{eq} = \|C^{eq}(x, u, p, t)\|_2^2 \quad \text{avec comme condition initiale } \dot{x}_{eq}(0)=0 \quad (2.17)$$

La contrainte (2.16) est équivalente à la contrainte sur la nouvelle variable au temps final :

$$x_{eq}(t_f) = 0 \quad (2.18)$$

Remarque : -1- La norme au carré peut également être utilisée comme fonction de pénalisation dans le cas de contraintes égalités.
-2- Le choix de la norme au carré (pour la fonction de pénalisation ou de la contrainte sur le temps final) introduit un gradient nul (par rapport aux variables de décision) à la solution. Ceci se traduit par une faible vitesse de convergence au voisinage de la solution Goh et Teo (1988). Vassiliadis et al (1994b) propose une alternative pour éviter ce problème.

Vu la difficulté de trouver une fonction de pénalisation pour prendre en compte les contraintes inégalités, une solution alternative consiste à introduire une variables d'écart pour les transformer en contraintes égalité :

$$C^{in}(x, u, p, t) + S^2 = 0 \quad (2.19)$$

Bien qu'attrayante, cette technique souffre d'un désavantage majeur. Dans la plus part des cas pratiques, ces contraintes inégalités dépendent uniquement des variables d'état $x(t)$. L'introduction de (2.19) dans le système EDA initial conduit à augmenter l'index du système. Une procédure de réduction d'index doit être alors mise en œuvre par différentiation de (2.19) et éventuellement de certaines équations du système initial (2.1) et par manipulations algébriques (Vassiliadis et al (1994b)).

Remarque : D'autres techniques approximent la satisfaction des contraintes grâce à l'introduction d'une fonction de pénalisation exacte (Bloss et al (1999)).

Quelles que soient les méthodes basées sur une approche séquentielle, elles ne peuvent pas prendre en compte les contraintes dues à des limitations sur les variables d'état (exemple : température maximum à ne pas dépasser, composition minimum etc.). En effet, les variables d'état n'interviennent pas dans les variables de décision du problème de programmation non linéaire et donc ces contraintes ne peuvent pas être prises en compte (bien évidemment les techniques présentées précédemment ne peuvent pas s'appliquer à ces contraintes particulières). Toutefois, Vassiliadis (1994b) et Serban et Petzold (2001) proposent une méthode pour remédier à cet inconvénient. Le principe est de discrétiser l'horizon de temps, elle sera exposée plus en détail dans le prochain chapitre. Feehery et Barton (1998) ont également développé une autre méthode pour inclure l'ensemble des contraintes inégalités sans complexifier le problème initial. Ils établissent un parallèle entre l'optimisation dynamique sous contraintes inégalités et la simulation des systèmes dynamiques hybrides. Lors de la simulation des systèmes dynamiques hybrides, l'intégrateur peut détecter des événements, et dans certains cas cette détection d'événement conduit à un changement de modèle (exemple : apparition d'une phase vapeur etc.). C'est sur l'aspect détection d'événement et changement de modèle que les auteurs ont basé la similitude entre les deux

approches. Durant l'intégration du système EDA, les contraintes inégalités sont parfois vérifiées et donc non actives, et parfois actives pour délimiter une région non admissible :

Parfois $C_m(x,u,p,t) < 0$ non actives

Parfois $C_m(x,u,p,t) = 0$ actives

L'algorithme proposé détecte automatiquement l'activation ou non des contraintes pendant la résolution du système EDA et des sensibilités (nécessaires pour la résolution du problème de programmation non linéaire). Cependant, leur activation entraîne l'ajout des contraintes égalités au système EDA initial ce qui risque d'augmenter l'index du système final (système initial avec les contraintes égalités) par rapport au système initial.

Les approches séquentielles ont été très utilisées dans des applications variées en Génie des procédés avec par exemple : Sargent et Sullivan (1979), Sorensen et al (1996) en distillation réactive, Serban et Petzold (2001) pour le transfert de chaleur, Vassiliadis et al (1994a et b) en réacteur batch, Mujtaba et Machietto (1997), Furlonge et al (1999) en distillation batch, Feehery et Barton (1999), Ishikawa et al (1997), Elgue(2002) pour le couplage réaction puis séparation...

IV.2 Approche simultanée

Dans l'approche séquentielle, l'intégration du système EDA (étape 3) est très consommatrice en temps calcul. Dans l'approche simultanée, l'intégration explicite de l'EDA est évitée par approximation des équations. La discrétisation concerne toujours les variables de contrôle mais elle est également étendue aux variables du système EDA (Cuthrell et Biegler (1987)). Au final, ces discrétisations génèrent un problème de programmation non linéaire de grande taille, souvent résolu par un algorithme de programmation quadratique successive. Cette approche simultanée est qualifiée d'approche à chemin irréalisable car le système EDA est résolu seulement au point solution du problème d'optimisation (et non pas à chaque itération comme dans l'approche séquentielle) ce qui évite des solutions intermédiaires qui n'existerait pas ou qui demanderait un effort d'intégration excessif. Le principe général de cette approche peut se résumer en quatre étapes :

- 1- Paramétrisation des variables de contrôle et d'état en utilisant un nombre fini de variables de décision (polynôme).
- 2- Discrétisation des équations différentielles (2.1). Ces deux étapes transforment le problème d'optimisation dynamique en un problème classique de programmation non linéaire.
- 3- Initialisation des variables de décision.
- 4- Résolution itérative grâce à une méthode de programmation non linéaire.

Pour être plus précis sur l'étape 2, la transformation du problème est réalisée grâce à une méthode de collocation. Dans ce cas, l'horizon de temps est divisé en éléments, à cause

d'éventuelles variations rapides des variables d'état. Sur chacun de ces sous intervalles, les variables de contrôle et d'état sont approximées par des polynômes. Les coefficients de ces polynômes sont ensuite, considérés comme des variables de décision pour le système en des points de collocation dans chaque intervalle de temps (Bathia et Biegler (1996)). L'amélioration de la précision requiert un grand nombre de points de collocation mais en contrepartie ce nombre de points ne doit pas être trop élevé car les polynômes de degrés élevés ont tendance à osciller (Tanartkit et Biegler (1995)). Souvent, sur chaque élément les équations du système EDA sont satisfaites en des points de collocation orthogonale. Toutefois, d'autres méthodes de collocations peuvent être utilisées Betts (2001) ou des différences finies Jockenhövel et al (2003). Dans la littérature, la représentation des polynômes peut être multiple : polynômes de Lagrange pour Curthrell et Biegler (1987), base des monômes Cervantes et Biegler (1998) (représentation recommandée de part son faible nombre de conditions et moins d'erreur d'arrondi).

A titre d'exemple, voici une formulation possible du problème, toujours avec objectif de minimiser la fonction objectif (les contraintes sur les variables d'état ne sont pas spécifiées pour alléger la présentation) :

L'horizon de temps est subdivisé en intervalle : $t_0 < t_1 < t_2 < \dots < t_N$

Les variables différentielles et les variables algébriques sont respectivement notées x_1 et x_2 . Comme énoncé précédemment, le problème d'optimisation (avec le système EDA comme contrainte) est converti en un problème de programmation linéaire par approximation des variables d'état et de contrôle par des familles de polynômes sur les sous intervalles de temps. La partie différentielle du système EDA peut être représentée à l'aide de la base des monômes (par exemple) :

$$x_1(t) = x_{1,i-1} + h_i \sum_{k_1=1}^q \chi_{k_1} \left(\frac{t - t_{i-1}}{h_i} \right) \frac{dx_1}{dt}_{i,k_1} \quad (2.20)$$

Où :

$x_{1,i-1}$: valeur des variables différentielles au début du sous intervalle i

h_i : longueur de l'élément i

$\frac{dx_1}{dt}_{i,k_1}$: valeur de la dérivée première sur le sous intervalle i au point de collocation k_1

χ_{k_1} : polynôme d'ordre q

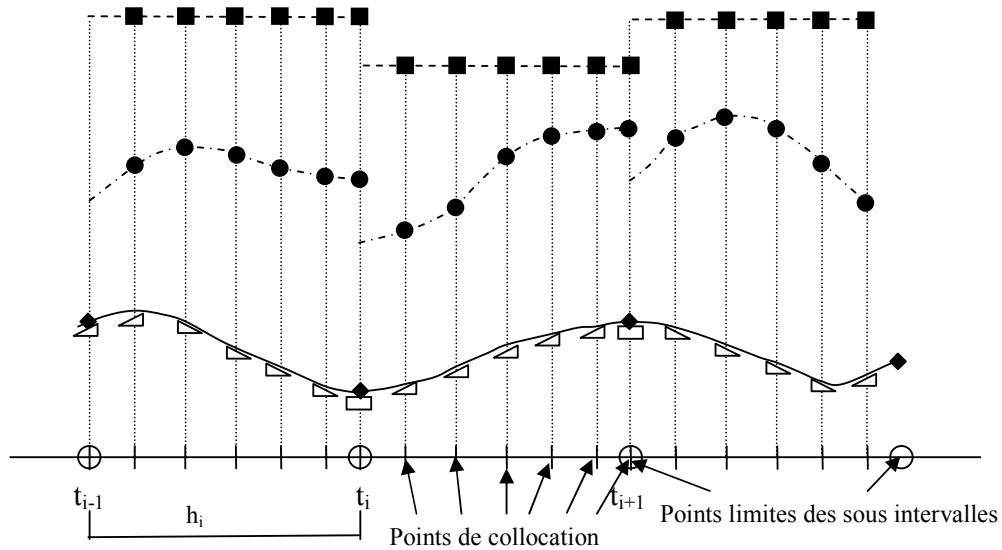
Les profils de contrôle et des variables algébriques peuvent, quant à eux, être approximés à l'aide de polynômes de Lagrange d'ordre q (par exemple) :

$$x_2(t) = \sum_{k_1=1}^q \vartheta_{k_1} \left(\frac{t - t_{i-1}}{h_i} \right) x_{2,i,k_1} \quad (2.21)$$

$$u(t) = \sum_{k=1}^q \vartheta_{k1} \left(\frac{t-t_{i-1}}{h_i} \right) u_{i,k1} \quad (2.22)$$

$x_{2,i,k1}$ et $u_{i,k1}$ représente les valeurs des variables de contrôles et algébriques, sur l'intervalle i au point de collocation k_1 .

Cette mise en forme peut se schématiser comme suit :



■ Fonction de contrôle aux points de collocation, ● Variables Algébriques aux points de collocation, ◆ Valeurs des variables différentielles aux points limites des sous intervalles.

Figure 2.2 : Représentation graphique de la méthode simultanée.

D'après cette figure 2.2, les variables différentielles sont continues sur l'ensemble de l'horizon de temps tandis que des discontinuités, aux points limites des sous intervalles (mais pas aux points de collocation), sont possibles sur les variables algébriques et de contrôles. La continuité aux points limites est imposée directement par l'équation (2.20), par contre pour les points de collocation cette continuité est forcée via des contraintes additionnelles.

De part la voie de transformation du problème d'optimisation dynamique, le problème de programmation linéaire généré est de grande taille. Il en résulte la mise en œuvre de méthodes efficaces pour résoudre ce problème. Avec le développement de méthodes de programmation quadratique successive ainsi que l'exploitation de la structure du problème (matrice creuse), de nombreuses techniques comme : approche du point intérieur, programmation quadratique successive avec réduction d'espace, gradients conjugués etc. ont permis de résoudre efficacement ce problème de programmation non linéaire (Cervantes et Biegler (1998), Cervantes et al (2000 a), Cervantes et Biegler (2000), Albuquerque et Biegler (1997), Vanderbei et Shanno (1999)).

Dans l'approche simultanée, les variables d'état font partie du problème d'optimisation et par conséquent les contraintes sont facilement prises en compte au niveau de la méthode de programmation non linéaire. De ce fait, l'inconvénient constaté sur les contraintes sur les variables d'état lors de l'approche séquentielle n'est plus présent. Pour plus de détails sur ces aspects ainsi que sur les approches simultanées et les stratégies efficaces de résolution, le lecteur peut se référer aux articles de synthèse suivant : Kameswaran et Biegler (2006), Biegler (à paraître), Biegler et al (2002).

Grâce à leur facilité à inclure les contraintes (notamment sur les variables d'état), ces méthodes ont largement été employées dans le domaine du génie des procédés : Cervantes et al (2000b et 2002), Jockenhövel et al (2003) sur un procédé complet, Raghunathan et al (2004) pour de la distillation, Lakshmanan et al (1999), Tamarkit et Biegler (1996 et 1997), Ruppen et al (1995) pour un réacteur batch, Albuquerque et al (1999) pour un modèle prédictif de contrôle non linéaire, Kameswaran et Biegler (2006) pour un problème de thermique, (Biegler à paraître) lors d'une cristallisation et d'une distillation réactive.

V Programmation Dynamique Itérative

La méthode de Programmation Dynamique Itérative (PDI) pour la résolution du problème de contrôle optimal se limite aux problèmes de petite taille. Cette méthode a considérablement augmenté son efficacité avec les travaux de Bojkov et Luus (1992), Dadebo et Mc Auley (1995), Mekarapiruk et Luss (1997). Bien que la méthode PDI soit bien plus lente que les méthodes basées sur la détermination du gradient de descente, elle est utilisable pour la vérification de résultats pour les problèmes de petite taille car elle peut éviter les minima locaux. Avec cette méthode la probabilité de chance d'atteindre l'optimum global est souvent élevée Dadebo et Mc Auley(1995), si le découpage sur les variables d'état est bien choisi. Le principe général de cette méthode est de diviser l'horizon de temps en P sous intervalles de même durée sur lesquels les variables de contrôle sont considérées constantes (les variables de contrôle sont donc constantes par morceaux). L'objectif est de déterminer sur chaque intervalle le vecteur u qui minimise la fonction de coût. Le dernier intervalle de discrétisation sert de point de départ avec un ensemble de valeurs pour le contrôle. Pour l'ensemble des variables d'état, le meilleur contrôle est retenu et l'algorithme procède par « rétro résolution » en répétant l'étape précédente jusqu'au premier intervalle. Une fois le point initial atteint l'ensemble du principe et repris avec un ensemble de valeurs de contrôle plus fin autour de la meilleure valeur retenue, ainsi la totalité du processus est réitéré. L'algorithme complet est détaillé dans les articles de Luus et Rosen (1991), Bojkov et Luus(1992)... La figure 2.3 illustre graphiquement la méthode PDI pour le vecteur de contrôle u réduit à un scalaire, méthode détaillée dans ce cas particulier, Dadebo et Mc Auley (1995).

Description de la méthode pour le cas particulier : une fonction de contrôle.

Pour appliquer la méthode PDI, l'intervalle de temps est subdivisé en P intervalles de même durée L puis un nombre de vecteurs d'état admissibles sur chaque intervalle doit être choisi. Pour se faire, deux voies sont possibles : générer une grille uniforme de variables d'état ou trouver ces points par intégration. Cette dernière est la plus communément employée. Il suffit de prendre N valeurs admissibles (constantes car la fonction de contrôle est supposée constante sur un intervalle de temps) pour le profil de contrôle dans une région r (région délimitant l'ensemble des contrôles admissibles), puis d'intégrer le système (2.1) et ainsi générer N trajectoires possibles pour les variables d'état, N=10 sur la figure 2.3. En stockant les valeurs de x aux bornes de chaque intervalle de temps, il en résulte un maillage de N vecteurs x pour chaque intervalle. Bien évidemment, le premier intervalle est un peu particulier car il ne contient que la condition initiale comme vecteur d'état.

Plus précisément, il ne s'agit pas de trouver N valeurs de u pour intégrer l'EDA (2.1) pour générer les trajectoires. En fait, il suffit d'en connaître une u^0 puis en tenant compte de l'amplitude de la région admissible r, les N-1 autres valeurs sont générées automatiquement à partir de cette valeur.

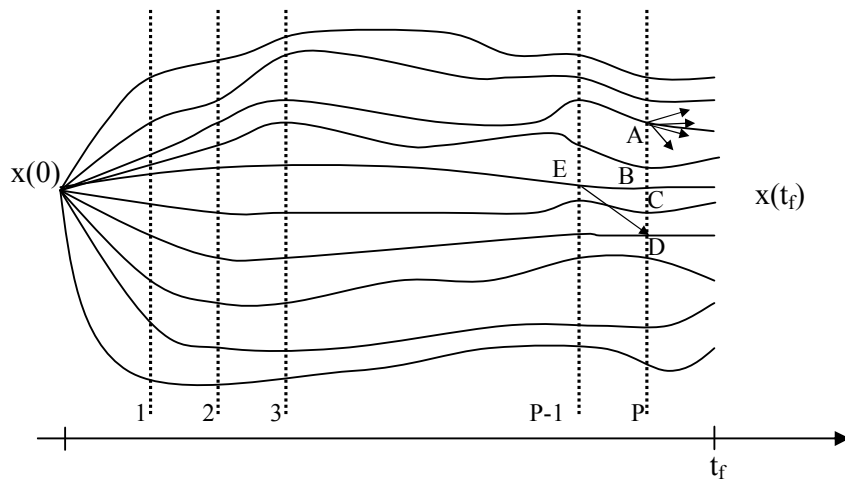


Figure 2.3 : Représentation Graphique de la méthode PDI (Dadebo (1995))

Une fois ces étapes préliminaires achevées, il faut s'intéresser au dernier intervalle de temps, P, dont le point initial est en t_f-L . Les N vecteurs d'état en t_f-L deviennent des conditions initiales pour une intégration sur le dernier intervalle. A partir de chaque vecteur d'état, la fonction de contrôle est susceptible de prendre M valeurs (comprises dans la région r). Par exemple, à partir du point A, le système est intégré avec les M valeurs de u, et ainsi il y a normalement M valeurs différentes pour le critère. Il suffit de stocker celle qui minimise ce dernier, et de l'utiliser comme valeur du contrôle pour toute trajectoire arrivant au point A (ou

en son voisinage immédiat). Cette procédure est répétée pour les points B, C ainsi que pour les 7 autres points dans l'exemple présenté. Donc au final pour chacun des points initiaux du dernier intervalle, le contrôle optimale est connu et stocké.

L'étape suivante s'intéresse à l'intervalle P-1, i.e. de t_{r-2L} à t_{r-L} . Pour chacun des vecteurs d'état, la technique précédente est employée mais non pas jusqu'à la fin de l'intervalle mais jusqu'au temps final. Par exemple au point E, pour une valeur du contrôle le système (2.1) est intégré jusqu'au point t_{r-L} et pour continuer jusqu'au point final t_f , il suffit d'utiliser le contrôle optimal du point le plus proche trouvé à l'étape précédente, ici D. Il faut répéter le processus pour chacune des M valeurs du contrôle et évaluer le coût pour trouver la trajectoire optimale issue de E allant jusqu'au temps final. Il faut faire de même pour chaque vecteur d'état du temps t_{r-2L} . Cette procédure de « rétro résolution » est répétée jusqu'au premier intervalle qui ne possède qu'un vecteur d'état $x(0)$. Et au final, lorsque les M intégrations possibles issues de $x(0)$ sont comparées sur la base de la fonction coût, la meilleure politique de contrôle est obtenue pour joindre $x(0)$ à $x(t_f)$, cette politique de contrôle jouera le rôle de u_0 pour la prochaine itération. Mais avant de passer à l'itération suivante, la taille de la région des contrôles admissibles r est réduite en lui appliquant un facteur de contraction : $r^{j+1} = \gamma^j r^j$ avec $0,5 \leq \gamma < 1$. Connaissant cette nouvelle région et le nouveau u_0 , il est facile de générer les N nouveaux vecteurs d'état. L'ensemble de cette procédure itérative est répétée et le résultat analysé (en général entre 20 et 30 itérations).

Par généralisation de l'approche précédente à m paramètres de contrôle, il faut alors comparer M^m trajectoires pour chaque point. Afin de réduire le temps calcul dû à l'explosion des combinaisons envisageables, Bojkov et Luus(1992) propose de choisir aléatoirement R valeurs pour les paramètres de contrôles dans l'ensemble M^m . Ils démontrent qu'une valeur maximum de $R=100$ peut être retenue, indépendamment de m . L'algorithme général de cette méthode est présenté en détail par Luus et Rosen (1991) et Dadebo et Mc Auley (1995) (présenté en Annexe 2). Plutôt que de considérer les paramètres de contrôle constants par palier, Luus (1993) propose une variation linéaire sur chaque intervalle de temps (mais dans ce cas $N=1$).

Cet algorithme fonctionne très bien lorsque le système initial ne comporte pas de contraintes d'état. La prise en compte de ces contraintes se fait par pénalisation de la fonction coût : ajout d'un terme à la fonction objectif pour pénaliser la violation des contraintes. Cet ajout est réalisé soit en introduisant une variable d'état pour chaque inégalité ; dont le but est de mesurer la violation de la contrainte au cours du temps (Mekarapiruk et Luus (1997)), soit en calculant l'éventuelle violation des contraintes en des points de temps donnés (Dadebo et Mc Auley (1995)).

Cette méthode a été très employée pour l'optimisation de réacteurs en régime dynamique : Rosen et Luus (1991), Luus et Rosen (1991), Dadebo et Mc Auley (1995), Bojkov et Luus(1992). Toutefois, ses applications en génie chimique ne se limitent pas aux réacteurs : Luus (1993) l'utilise pour un fermenteur fed batch, pour une réaction pour Bojkov et Luus (1994), une colonne à distiller (Fikar et al (1998)), écoulement dans un réacteur tubulaire pour Mekarapiruk et Luus (1997)...

VI Conclusion

La figure 2.4 résume un grand nombre des approches possibles pour résoudre le problème d'optimisation dynamique (souvent les plus utilisées, c'est pourquoi la méthode PDI n'apparaît pas).

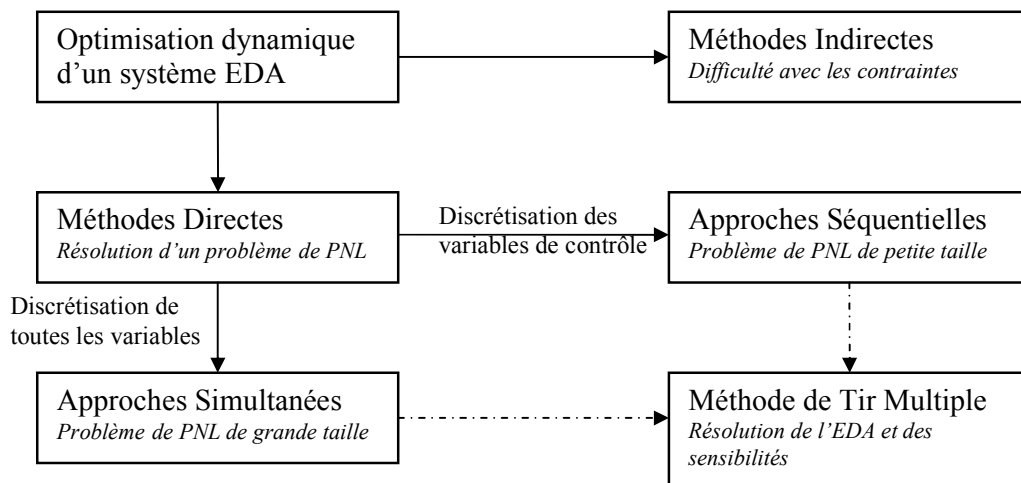


Figure 2.4 : Méthodes pour l'optimisation dynamique

Au départ, les premières stratégies pour résoudre les problèmes d'optimisation dynamique utilisaient des méthodes indirectes basées sur l'application du principe du maximum de Pontryagin. Pour obtenir des solutions avec ces méthodes, une attention particulière est à apporter aux conditions limites. Dans la majorité des cas, les variables d'état possèdent des conditions initiales et les variables de l'adjoint des conditions finales. Ce problème de valeurs limites aux deux bouts, peut être résolu avec la méthode de tir simple ou la méthode de tir multiple (voire d'autres mais elles sont moins usitées). Cependant, lorsqu'il faut prendre en considération des contraintes inégalités sur les variables d'état, la résolution devient plus complexe : notamment il devient très difficile de trouver des valeurs initiales cohérentes pour les variables d'état et de l'adjoint.

Les méthodes basées sur la résolution d'un problème de programmation non linéaire peuvent se scinder en deux catégories principales : les méthodes séquentielles et les méthodes simultanées. Pour la première catégorie, seules les variables de contrôle sont discrétisées (souvent sous forme polynômiale). Dans ce cas l'optimisation est réalisée par rapport aux coefficients des polynômes. Avec des conditions initiales cohérentes couplées à un ensemble de paramètres de contrôle, l'intégration du système EDA est incluse dans la résolution du problème de programmation non linéaire. A chaque itération de l'optimisation, les coefficients des polynômes de contrôle sont recalculés et le système EDA intégré avec ces nouvelles valeurs. Ces méthodes séquentielles sont relativement faciles à mettre en œuvre et donnent des résultats probants (surtout si l'intégrateur EDA et la méthode d'optimisation sont robustes). Par contre, l'intégration systématique du système EDA à chaque itération peut devenir très consommatrice en temps calcul, surtout dans le cas de systèmes de grande taille. De plus, la prise en compte de contraintes inégalités sur les variables d'état est réalisée de façon approximative grâce à des fonctions de pénalisation.

Pour les approches simultanées, la discrétisation concerne à la fois les variables de contrôle et les variables d'état. Par conséquent, ces approches conduisent à des problèmes de programmation linéaire de grande taille qui demandent alors des méthodes de résolution dédiées et efficaces. De par la discrétisation des variables d'état, la solution du système EDA est directement couplée au problème d'optimisation. Au final, le système EDA est résolu seulement une fois, au point optimal, évitant des coûts de calcul importants ainsi que des solutions intermédiaires qui n'existeraient pas. Ces méthodes sont robustes et peuvent passer outre des instabilités sur les entrées, ce qui est utile pour certains systèmes. Enfin, la prise en compte des contraintes sur les variables d'état et sur le contrôle est réalisée au même niveau de discrétisation que les variables du système EDA, ce qui en facilite l'incorporation.

Sur la figure 2.4, une dernière catégorie de méthodes apparaît : méthodes directes à tir multiple. Elles peuvent être considérées comme une méthode intermédiaire entre les deux précédentes. Elles sont souvent classées avec les approches simultanées mais elles possèdent de nombreux avantages hérités des méthodes séquentielles. Dans leur principe, ces méthodes partitionnent l'horizon de temps en sous intervalles où le système EDA est intégré (séparément sur chaque intervalle). Par contre les variables de contrôle sont traitées de façon identique à l'approche séquentielle. Le chapitre suivant dresse une analyse détaillée de cette méthode.

CHAPITRE 3

Présentation de la Méthode

Nomenclature du Chapitre 3

- C_0 : fonction de coût (fonction objectif)
 C : vecteur des contraintes sur les variables d'état
 C_i : contraintes de continuité
 C_p : vecteur des contraintes sur les variables d'état aux points de subdivision de l'intervalle de temps
 F : système EDA global ; système initial avec les équations de sensibilités
 f : système des équations différentielles et algébriques
 f^l : fonction à minimiser dans la méthode PQS
 g : vecteur des contraintes inégalités pour la méthode PQS
 h : vecteur des contraintes égalités pour la méthode PQS
 h_0 : fonction pour calculer la partie de Lagrange de la fonction de coût
 h_p : pas d'intégration
 J_p : jacobien par rapport aux paramètres
 J_u : jacobien par rapport aux fonctions de contrôles
 J_X : jacobien par rapport au vecteur des variables d'état X aux points de subdivision du temps
 J^{*-1} : la matrice jacobienne factorisée d'une itération précédente
 L : Lagrangien
 LI_{uj} : limite inférieure pour la fonction de contrôle j
 LI_{duj} : limite inférieure pour la dérivée de la fonction de contrôle j
 LS_{uj} : limite supérieure pour la fonction de contrôle j
 LS_{duj} : limite supérieure pour la dérivée de la fonction de contrôle j
 M : opérateur dynamique du système global
 m : opérateur dynamique du système initial
 N_c : nombre de sous intervalles de contrôle sur un sous intervalle de temps
 N_m : nombre de sous intervalles de subdivision du temps
 N_u : dimension du vecteur des fonctions de contrôle
 p : vecteur des paramètres opératoires
 p' : vecteur des paramètres opératoires de p et des coefficients des polynômes de contrôle
 q : ordre des monômes
 r_i : $i^{\text{ème}}$ résidu
 s_i : vecteur des sensibilités par rapport au paramètre p_i
 \dot{s}_i : vecteur des dérivées de s_i
 t : variable indépendante d'intégration
 t^* : variable d'intégration adimensionnelle (ici le temps)
 u : vecteur des fonctions de contrôle
 \dot{u} : vecteur des dérivées premières des fonctions de contrôle

$u_{j,i}^{k,q}$: coefficient du monôme d'ordre q de la fonction de contrôle j sur le sous intervalle de temps i, et sur le sous intervalle de contrôle k

X_i : valeur du vecteur des variables d'état x au point de subdivision du temps i (variable de décision qui assure la continuité)

x : vecteur des variables d'état (inconnues) de dimension n

\dot{x} : vecteur des dérivées premières des variables d'état

y : variables d'optimisation pour la méthode PQS

y* : solution pour le problème d'optimisation

α^k : longueur du pas d'optimisation

β : constante dépendant de l'ordre de la méthode de Gear

Δt_i : longueur d'un sous intervalle de contrôle

Δx^{k-1} : pas dans le correcteur (Newton Raphson)

δ_i : scalaire réel de faible valeur (calcul de dérivées numériques)

δy^k : pas de l'optimisation

$\phi(t)$: valeur de la fonction objectif à un instant donné

$\dot{\phi}(t)$: dérivée de $\phi(t)$

ψ : polynôme fonction des variables des pas précédents dans la méthode de Gear

λ, μ : paramètres de Lagrange et Kuhn Tucker

Symboles :

∇ : gradient

∇^2 : Hessien

Indices/exposants :

0 : condition initiale

f : condition finale

i : point de subdivision de l'intervalle de temps

k : itération lors de la présentation de la méthode PQS

T : transposée

* : point optimal

I Introduction

L'objectif de ce travail est d'ajouter un élément à la plateforme PrODHyS et plus précisément au niveau du noyau numérique. Cet élément devra permettre de réaliser l'optimisation de système dynamique (résoudre le problème de contrôle optimal de façon générale et quel que soit le problème). Le chapitre précédent donne un panel de méthodes pour résoudre ce problème. Dans le domaine du génie des procédés, la présence des contraintes inégalités dans de nombreux modèles, oriente le choix vers les méthodes directes. Dans cette catégorie de méthodes, deux approches principales se distinguent, la différence majeure entre ces approches provient du niveau de discrétisation. De ce fait, l'approche simultanée permet la prise en compte de contraintes sur les variables d'état, mais en contrepartie le système à résoudre est de très grande taille (discrétisation des variables d'état et de contrôle). Vu le domaine d'application retenu, il est primordial d'inclure les contraintes sur les variables d'état (contraintes de température, de pression, de composition ...) mais sans trop augmenter la taille du problème car les modèles générés dans ce domaine sont régulièrement de taille conséquente. Afin de satisfaire les contraintes sur la méthode de

résolution, une approche hybride entre les approches simultanée et séquentielle est présentée dans ce chapitre. Cette méthode nommée : méthode directe à tir multiple est basée sur les travaux de Bock et Platt (1984) repris et améliorés par Serban et Petzold (2001) et Leineweber et al (2003 a et b). L'idée principale de cette approche est de découper l'horizon de temps en sous intervalle afin de réaliser l'intégration. Excepté le premier sous intervalle, les conditions initiales de chaque sous intervalle (sur les variables d'état) sont incluses dans les variables de décision. Cependant, ce découpage nécessite l'ajout de contraintes de continuité pour les variables d'état traduisant le fait que l'état initial de chaque sous intervalle correspond à l'état final du sous intervalle précédent. Bien que le système EDA soit intégré précisément à chaque itération, cette méthode est à chemin irréalisable tout comme l'approche simultanée. Pour la méthode directe à tir multiple, le traitement des variables de contrôle est identique à celui réalisé dans l'approche simultanée, à savoir une paramétrisation de ces variables. Au final cette paramétrisation et ce découpage conduisent à un problème de programmation non linéaire, résolu dans la partie optimisation.

II Présentation Générale de la méthode

Avant de présenter la méthode dans ses moindres détails, cette partie s'attache à mettre en évidence les principes de cette dernière. Tout d'abord, en présentant comment le problème est abordé : avec une discrétisation de l'horizon de temps. Discrétisation qui n'est pas sans effet sur le problème de contrôle optimal puisqu'elle a pour conséquence d'introduire des variables et contraintes supplémentaires. Au final, le problème se résume à un problème de programmation non linéaire avec intégration du système EDA à chaque itération. Ce problème de programmation non linéaire est résolu par une méthode de programmation quadratique successive, brièvement détaillée dans le second paragraphe de cette partie.

II.1 Discrétisation de l'horizon de temps

Le point de départ est la résolution du problème de contrôle optimal dont la formulation est présentée au chapitre précédent et rappelée ici :

Soit le système EDA :

$$f\left(\dot{x}, x, p, u\right) = 0 \quad x(t=0) = x_0 \quad (3.1)$$

Le vecteur de paramètres p ainsi que celui de fonction de contrôle $u(t)$ sont déterminés de façon à ce que la fonction objectif soit minimisée :

$$C_o(t_f, u) = \int_{t_0}^{t_f} h_o(t, x(t), p, u(t)) dt \quad (3.2)$$

Tout en tenant compte des contraintes du problème :

$$C(t, x(t), p, u(t)) \leq 0 \quad (3.3)$$

La fonction de contrôle optimale $u^*(t)$ est supposée continue sur l'horizon de temps. Comme dans toute méthode directe, les fonctions de contrôles $u(t)$ sont paramétrées. Afin de réduire la dimension de l'espace de recherche de $u(t)$, sa représentation est sous forme polynomiale par morceau sur $[t_0 ; t_f]$, ainsi la détermination des coefficients est réalisée dans l'optimisation. Par la suite et par souci de simplification de la présentation, le vecteur p' englobera les paramètres et les coefficients des polynômes (p' est de dimension M) et ainsi supprimer $u(t)$ dans ce paragraphe.

Comme énoncé précédemment, dans cette méthode l'horizon de temps est subdivisé en sous intervalle $[t_i ; t_{i+1}]$ sur lesquels le système EDA est résolu. Sur ces sous intervalles des variables intermédiaires sont introduites dans le but d'assurer la continuité sur les variables d'état. Cette continuité se traduit par l'égalité entre l'état initial d'un sous intervalle et l'état final du sous intervalle précédent. Sur un sous intervalle quelconque, soit X_i la condition initiale pour l'intégration, ainsi sur ce sous intervalle, la solution du système EDA à l'instant t est notée $x(t, t_i, X_i, p')$. Dans ces conditions, les contraintes de continuité entre les sous intervalles s'écrivent :

$$C_i : X_{i+1} - x(t_{i+1}, t_i, X_i, p') = 0 \quad (3.4)$$

Avec ce découpage de l'horizon de temps, les contraintes du problème (3.3) doivent être satisfaites aux limites des intervalles de tir. Soit :

$$C_p(X_i, p') = C_p(t_i, X_i, p') \leq 0 \quad (3.5)$$

La discrétisation de l'horizon de temps a l'avantage de pouvoir facilement inclure les contraintes sur les variables d'état, ce qui était impossible dans le cas d'une approche séquentielle classique (cette dernière pouvant être qualifiée de méthode directe de tir simple puisque l'intervalle de temps n'est pas subdivisé).

Toutefois, imposer les contraintes (3.5) aux points de discrétisation ne garantit pas le respect de ces contraintes sur l'ensemble de l'horizon de temps. En effet, durant l'intégration sur chaque sous intervalle rien n'empêche de passer outre ces contraintes car seules les bornes de l'intervalle sont incluses dans les variables de décision de l'optimisation et par conséquent soumises aux contraintes. Une façon de remédier à ce problème serait de multiplier les points de discrétisation afin d'avoir une discrétisation très fine et donc éviter la violation des contraintes. La contrepartie de la multiplication de ces points, est l'augmentation de la taille du système de programmation non linéaire à résoudre. Une alternative consiste à éviter une discrétisation trop fine en ajoutant à la fonction objectif, des fonctions de pénalisation qui traduisent le non respect des contraintes (confère le chapitre précédent).

La méthode directe à tir multiple se résume par la figure suivante :

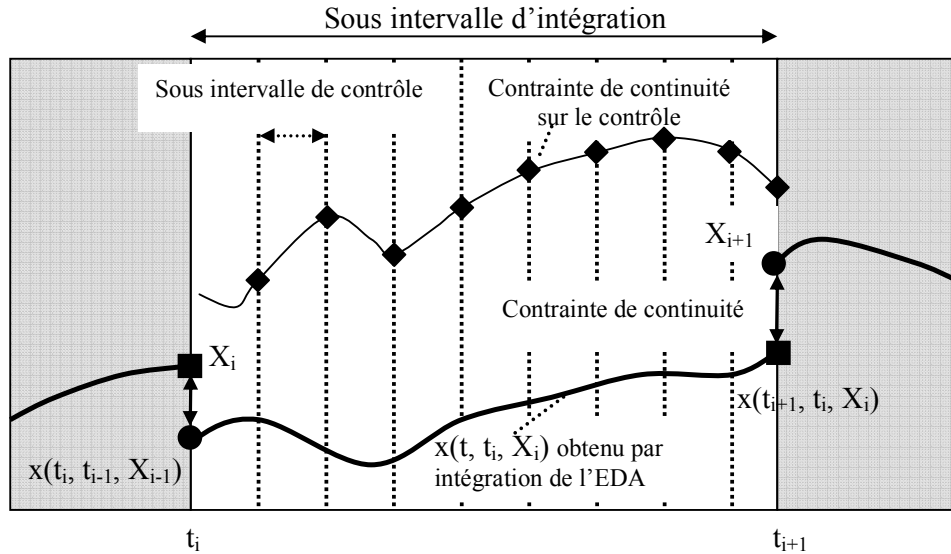


Figure 3.1 : Présentation graphique de la méthode

Au final, le système EDA est intégré sur chaque sous intervalle, à chaque pas de l'optimisation. De plus, cette méthode directe à tir multiple possède l'avantage d'être parallélisable, en effet chaque sous intervalle est indépendant ce qui les rend intégrables simultanément.

Concernant la fonction objectif, en appliquant la remarque du chapitre précédent (paragraphe II.1), il est possible de se ramener à un problème de Mayer :

$$\text{Soit } \phi(t) = \int_{t_0}^t h_o(t, x(t), p') dt \quad (3.6)$$

qui satisfasse

$$\dot{\phi}(t) = h_o(t, x(t), p') \quad \text{et} \quad \phi(t_0) = 0 \quad (3.7)$$

Ainsi, on peut introduire une nouvelle équation et une nouvelle variable au système EDA initial (3.1).

Après discrétisation de l'intervalle de temps, le problème de contrôle optimal (discrétisé) se met sous la forme :

$$\text{Min}_{X_2, \dots, X_m, p'} \phi(t_f) \quad (3.8)$$

Soumis aux contraintes :

$$C_i(X_{i+1}, X_i, p') = 0 \quad (3.9)$$

$$C_o(X_i, p') \leq 0 \quad (3.10)$$

Une façon de résoudre le problème de programmation non linéaire ainsi obtenu, est d'employer une méthode de programmation quadratique successive dont le principe est brièvement présenté dans le paragraphe suivant.

II.2 Méthode de Programmation Quadratique Successive

Présentation

De nos jours ces méthodes sont les plus efficaces et les plus utilisées pour résoudre des problèmes de programmation non linéaire, comme le démontre Kao (1998). Dans son principe cette méthode repose sur la résolution itérative (successive) d'une séquence de sous problèmes obtenus par linéarisation des contraintes non linéaires et par approximation quadratique du lagrangien. L'avantage principal de cette méthode réside dans le fait qu'elle est d'ordre 2. Pour cette méthode, il existe deux alternatives :

- Tous les points calculés lors de l'optimisation vérifient l'ensemble des contraintes : approche à chemin réalisable. En cas d'arrêt prématuré de la recherche cette approche est avantageuse mais elle traite que des cas particuliers comme les contraintes linéaires.
- La seconde alternative autorise le non respect de certaines contraintes au cours de l'optimisation, dans le but d'accélérer la convergence. En contre partie, en cas d'échec de la procédure, le point d'arrêt n'a pas forcément de signification, ce qui constitue un inconvénient car seul le point de convergence est acceptable.

Etude théorique

De manière générale, le problème se formule comme suit :

$$\text{Min } f^1(y) \tag{3.11}$$

$$\text{Avec : } g(y) \leq 0 \text{ vecteur des contraintes inégalité} \tag{3.12}$$

$$h(y) = 0 \text{ vecteur des contraintes égalité} \tag{3.13}$$

Pour que le problème soit résoluble, il faut que f^1 et g soient continues et deux fois dérivables sur le domaine. La résolution de ce problème sous contraintes passe par l'écriture de la fonction Lagrangienne associée :

$$L(y, \lambda, \mu) = f^1(y) + \lambda^T g(y) + \mu^T h(y) \tag{3.14}$$

λ et μ représentent respectivement les vecteurs des paramètres de Lagrange et Kuhn Tucker. Grâce à l'approximation quadratique de la fonction lagrangienne, à l'aide d'un développement de Taylor à l'ordre 2 (sur une direction admissible) et à la linéarisation des contraintes, à partir d'un point de départ donné y^k , l'objectif est alors, de déterminer le pas δy^k pour passer à l'itération suivante. δy^k est un minimum de l'approximation quadratique :

$$\text{Min} \left[\nabla^T f^1(y^k) \delta y^k + \frac{1}{2} \delta y^{kT} \nabla^2 L(y^k, \lambda, \mu) \delta y^k \right] \tag{3.15}$$

$$\text{Avec : } g(y^{k+1}) = g(y^k) + \nabla^T g(y^k) \delta y^k \leq 0 \quad (3.16)$$

$$h(y^{k+1}) = h(y^k) + \nabla^T h(y^k) \delta y^k = 0 \quad (3.17)$$

Remarque : La fonction à minimiser (3.15) n'est pas l'approximation quadratique du lagrangien au point y^k , en effet il manque le terme $f^l(y^k)$. Or ce dernier étant constant il peut être supprimé de la formulation.

Le pas δy^k , à lui seul, ne suffit pas pour calculer le nouveau point y^{k+1} , une recherche est effectuée le long du pas pour déterminer la longueur de ce dernier : α^k . α^k est un facteur de relaxation, introduit pour compenser l'approximation effectuée sur la fonction objectif et sur les contraintes. Le processus de résolution des problèmes quadratiques est répété jusqu'à l'obtention des conditions d'optimalité. Minoux (1983) démontre que la solution y^* est obtenue si les conditions suivantes sont vérifiées :

$$\left\{ \begin{array}{l} \nabla L(y^*, \lambda, \mu) = 0 \\ g(y^*) \leq 0 \\ h(y^*) = 0 \\ \mu \geq 0 \text{ et } \mu_j g_j(y^*) = 0 \quad \forall j \end{array} \right. \quad (3.18)$$

Ces conditions sont également appelées conditions de Kuhn Tucker ; conditions nécessaires d'optimalité du 1^{er} ordre. Toutefois, elles ne suffisent pas à garantir l'optimalité du point solution. Pour en assurer l'optimalité, les conditions du 2nd ordre doivent également être satisfaites. Ces dernières se traduisent mathématiquement par la matrice Hessienne du système, $\nabla^2 L(y^*, \lambda, \mu)$, qui doit être une matrice définie positive. Dans la majorité des problèmes, il est très difficile d'obtenir une valeur analytique de cette matrice (notamment au point solution), c'est pourquoi la plupart du temps les conditions de Kuhn Tucker du 1^{er} ordre suffisent pour estimer la qualité du point solution. La nécessité de calculer la matrice Hessienne ne se limite pas aux conditions d'optimalité, elle est également utile pour résoudre le sous problème quadratique. L'évaluation de cette matrice est souvent rédhibitoire à cause du temps calcul très coûteux. Par conséquent, dans l'implémentation de la méthode, la matrice Hessienne est généralement approchée par des procédures dont la plus usitée est la méthode de Broyden- Fletcher-Goldfarb-Shanno (BFGS).

L'algorithme général de la méthode de programmation quadratique successive se résume comme suit :

- 1- Initialisation du problème : Matrice Hessienne (identité), Evaluation de la fonction objectif et des contraintes

- 2- Evaluation des gradients
- 3- Formulation du sous problème quadratique
- 4- Vérification des conditions d'arrêt : si c'est le cas, la solution est trouvée, sinon
- 5- Estimation de la matrice Hessienne
- 6- Estimation des paramètres de Lagrange et Kuhn Tucker
- 7- Evaluation du pas de descente δy^k
- 8- Evaluation de la longueur du pas α^k
- 9- Calcul du nouveau point $y^{k+1} = y^k + \delta y^k$
- 10- Evaluation de la fonction objectif et des contraintes
- 11- Retour en 2

Le point 2 est une étape primordiale dans cet algorithme. En effet, il requiert le calcul des gradients du critère et des contraintes. Le calcul des gradients est crucial car ils déterminent la direction de descente et ils interviennent dans les conditions d'arrêt. Par conséquent, une attention particulière leur est portée dans la partie suivante.

Remarque : L'optimiseur utilisé est celui créé par L. T. Biegler et ses collaborateurs à l'université de STANFORD dans les années 80 Biegler (1985) et légèrement modifié et adapté par la suite. De plus, il est facilement interfaçable avec l'intégrateur.

III Résolution Numérique

Après cette présentation de la méthode dans ses grands axes, les points importants sont détaillés dans cette partie. Tout d'abord, le paragraphe précédent démontre qu'au final un problème de programmation non linéaire doit être résolu. La mise en œuvre de la méthode retenue pour le résoudre nécessite la connaissance des gradients par rapport aux variables d'action. Les équations de sensibilité sont une voie pour obtenir ces informations. Ces équations, couplées au système EDA initial forment le système EDA global à intégrer à chaque pas d'optimisation. Le second paragraphe de cette partie est focalisé sur la stratégie de résolution du système EDA global. Dans les deux derniers paragraphes, des aspects plus spécifiques à la méthode sont abordés. Dans un premier temps, la paramétrisation du vecteur des fonctions de contrôle, avec le découpage de ces fonctions sur chaque sous intervalle d'intégration, suivi ensuite, par la prise en compte des contraintes (issues du problème initial et générées par la méthode) pour le problème de programmation non linéaire.

III.1 Analyse des sensibilités

Les gradients précités peuvent être calculés à l'aide des équations de sensibilité issues du système EDA initial (3.1). L'analyse des sensibilités consiste à déterminer la dérivée des

variables du modèle initial (3.1) par rapport à chaque paramètre du vecteur p . Cette analyse des sensibilités est une information importante dans de nombreuses applications scientifiques et industrielles comme par exemple : l'estimation paramétrique, l'optimisation, la simplification de modèle, l'étude de sensibilités... Dans le cas présent, les problèmes de contrôle optimal peuvent nécessiter l'information sur les sensibilités pour un nombre important de paramètres. De plus, comme il a été mentionné dans l'introduction générale, il est fréquent d'avoir des modèles de grande taille en génie des procédés. Par conséquent, avec la méthode de contrôle optimal retenue, une méthode de calcul précise, peu coûteuse en temps calcul est primordiale pour estimer les sensibilités, vu leur impact sur le résultat.

Pour le calcul des sensibilités dans le cas de systèmes de grandes tailles, deux stratégies principales sont applicables : le calcul avant et le calcul inverse (adjoint). La première approche est particulièrement efficace pour le calcul de sensibilités pour des problèmes avec un grand nombre variables avec un nombre plus faible de paramètres. Au contraire, le calcul inverse, lui est avantageux lors du calcul des sensibilités pour une ou quelques variables (faible nombre) par rapport à un grand nombre de paramètres. Bien que la complémentarité des deux approches soit évidente, seule la première est retenue et présentée dans cette partie. Pour ce qui est de la seconde approche des détails sont donnés dans les articles suivants : Petzold et al (2000), Cao et al (2003), Li et Petzold (2003), Cao et al(2002), Petzold et al (2006).

Les équations de sensibilité

Le système EDA initial est sous la forme :

$$f(t, x, \dot{x}, p) = 0 \text{ avec } x(t_0) = x_0 \quad (3.19)$$

n_x est la dimension des variables et n_p celle des paramètres. Compte tenu de la définition des sensibilités, ceci engendre $n_x * n_p$ équations supplémentaires, de sensibilité.

Les sensibilités sont calculées via l'intégration d'équations générées par la dérivation de l'équation précédente par rapport aux paramètres :

$$\frac{\partial f}{\partial x} s_i + \frac{\partial f}{\partial \dot{x}} \dot{s}_i + \frac{\partial f}{\partial p_i} = 0 \quad (3.20)$$

$$s_i = \frac{\partial x}{\partial p_i} \quad \dot{s}_i = \frac{\partial \dot{x}}{\partial p_i} \quad i=1, \dots, n_p \quad (3.21)$$

A titre d'exemple, soit le système EDA suivant (tiré de l'article de Maly et Petzold (1996)) :

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} -(p_1 + p_3)x_1^2 \\ p_1x_1^2 - p_2x_2 \end{pmatrix}, \quad \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (3.22)$$

Ce système contient 2 variables d'état et 3 paramètres, soit au final 6 équations de sensibilités. Dans ce cas très simple, il est aisé de calculer analytiquement ces équations :

$$\begin{pmatrix} \dot{x}_1 + (p_1 + p_3)x_1^2 \\ \dot{x}_2 - p_1x_1^2 + p_2x_2 \\ \dot{s}_1 + 2s_1(p_1 + p_3)x_1 + x_1^2 \\ -2x_1p_1s_1 + p_2s_2 + \dot{s}_2 - x_1^2 \\ 2x_1(p_1 + p_3)s_3 + \dot{s}_3 \\ -2x_1p_1s_3 + p_2s_4 + \dot{s}_4 + x_2 \\ 2x_1(p_1 + p_3)s_5 + \dot{s}_5 + x_1^2 \\ -2x_1p_1s_5 + p_2s_6 + \dot{s}_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ avec } \begin{pmatrix} x_1(0) \\ x_2(0) \\ s_1(0) \\ s_2(0) \\ s_3(0) \\ s_4(0) \\ s_5(0) \\ s_6(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad (3.23)$$

Pour la plupart des modèles, la génération des équations de sensibilité n'est pas aussi triviale que dans le cas précédent. En génie des procédés, la forte non linéarité des équations (due à la thermodynamique, réaction, ...), la formulation implicite de certains termes, la grande taille des systèmes, ... sont autant d'inconvénients qui compliquent énormément la génération analytique des équations de sensibilité. Il devient alors fort intéressant d'explorer l'alternative numérique pour les générer. La résolution du système initial (sans sensibilités) par la méthode de Gear demande l'opérateur dynamique et donc par conséquent les termes : $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial \dot{x}}$

(paragraphe III.2). Ces deux termes sont calculés soit analytiquement, soit numériquement (choix laisser à l'utilisateur dans l'intégrateur DISCO). Outre ces deux termes (par conséquent déjà connus), il faut une information additionnelle pour calculer les résidus des équations de

sensibilité : $\frac{\partial f}{\partial p_i}$.

Il existe trois méthodes pour évaluer les résidus des équations de sensibilité. Comme on pourra le voir par la suite, le choix de la méthode peut affecter significativement le coût de la résolution.

Approche n°1 : Evaluation Analytique

Lorsqu'il est aisé d'évaluer analytiquement le terme $\frac{\partial f}{\partial p_i}$, le calcul des résidus des équations de sensibilité se réduit alors au produit de 2 matrices et la somme de 2 vecteurs :

$$r_i = A \begin{bmatrix} \dot{s}_i \\ s_i \end{bmatrix} + \frac{\partial f}{\partial p_i} \text{ avec } A = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial x} \end{bmatrix} \quad (3.24)$$

La matrice A est calculée en mettant à jour le Jacobien du système EDA initial (3.1).

L'avantage de calculer l'ensemble des termes analytiquement est indéniablement la précision des calculs des résidus. Cette méthode est moins coûteuse en temps calcul que les autres approches décrites ci-dessous. Malheureusement, elle reste limitée à un nombre restreint de problèmes.

Approche n°2 : Différences finies

L'évaluation analytique du terme $\frac{\partial f}{\partial p_i}$, n'est pas toujours possible, toutefois le calcul des résidus des équations de sensibilité est tout de même réalisable. Ce calcul des résidus nécessite alors une évaluation de ce terme via des différences finies, par exemple avec des différences finies du 1^{er} ordre donnant ainsi l'équation suivante pour :

$$r_i = A \begin{bmatrix} \dot{s}_i \\ s_i \end{bmatrix} + \frac{f(t, x, \dot{x}, p + \delta_i e_i) - f(t, x, \dot{x}, p_i)}{\delta_i} \quad (3.25)$$

δ_i : scalaire de faible valeur

e_i : i^{ème} vecteur canonique

Cette approche possède tous les inconvénients que peut apporter toute estimation de dérivées par différences finies : temps calcul, précision...surtout si les termes de la matrice A sont également évalués par différences finies.

Approche n°3 : Dérivées Directionnelles

Le point de départ de cette approche consiste à réaliser un développement de Taylor de f en x :

$$f\left(t, x + \delta_i s_i, x + \delta_i s_i, p + \delta_i e_i\right) = f\left(t, x, x, p\right) + \delta_i \frac{\partial f}{\partial x} s_i + \delta_i \frac{\partial f}{\partial x} s_i + \delta_i \frac{\partial f}{\partial p_i} + 0(\delta_i) \quad (3.26)$$

$$\Leftrightarrow \frac{f\left(t, x + \delta_i s_i, x + \delta_i s_i, p + \delta_i e_i\right) - f\left(t, x, x, p\right)}{\delta_i} = \frac{\partial f}{\partial x} s_i + \frac{\partial f}{\partial x} s_i + \frac{\partial f}{\partial p_i} + 0(\delta_i) \quad (3.27)$$

Par conséquent les résidus sont approchés par la formule suivante :

$$r_i = \frac{f\left(t, x + \delta_i s_i, x + \delta_i s_i, p + \delta_i e_i\right) - f\left(t, x, x, p\right)}{\delta_i} \quad (3.28)$$

Cette approche est qualifiée de méthode à dérivées directionnelles. Bien évidemment, le choix d'une valeur appropriée pour δ_i est crucial afin de maintenir des erreurs d'arrondi et de troncature acceptables. Pour une discussion plus poussée sur le choix de ce paramètre, le lecteur peut examiner les travaux de Maly et Petzold (1996).

Avec l'équation (3.28) l'erreur commise sur le $i^{\text{ème}}$ sensibilité est de l'ordre $0(\delta_i)$. Toutefois, il est possible d'avoir une erreur de l'ordre $0(\delta_i^2)$ via des différences finies centrées du 2nd ordre. Avec cette approche, les équations de sensibilité sont générées en même temps que les résidus du système initial via n_p (ou $2*n_p$) appels supplémentaires au sous programme calculant les résidus du système initial.

Il est intéressant de noter que l'approche par dérivées directionnelles est moins couteuse en temps calcul que l'approche n°2. Toutefois, sur ce critère, il est préférable de calculer analytiquement les résidus comme le démontre l'étude de Feehery et al (1997). Dans le cas où

les termes $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial p}$ sont calculés par différences finies, ce qui est souvent le cas, alors il

vaut mieux utiliser les dérivées directionnelles afin d'éviter l'introduction d'erreurs importantes.

Au final, dans notre code, l'utilisateur a le choix entre entrer les dérivées analytiques et l'approche 3 si ces dernières ne sont pas facilement disponibles.

III.2 Méthode d'intégration du système global

Méthode de Gear

Dans le cadre du système EDA initial (seul), la stratégie de résolution utilisée par l'intégrateur DISCo est basée sur le méthode Gear (Gear 1971) : il s'agit d'un algorithme du type prédicteur-correcteur basé sur la différentiation rétrograde. Cette méthode est succinctement décrite :

Soit un point (t_n, x_n, \dot{x}_n) connu lors de l'intégration. L'idée est de déterminer le point suivant x_{n+1} .

Dans un premier temps la dérivée est approximée :

$$\frac{\partial x_n}{\partial t} = \frac{x_{n+1} - \psi}{h_p \beta} \Leftrightarrow x_{n+1} = \psi + h_p \beta \dot{x}_n \quad (3.29)$$

ψ : polynôme fonction des variables aux pas précédents

h_p : pas d'intégration

β : constante dépendant de l'ordre de la méthode

Remarque : $x_n = \psi, \beta = 1$, soit Gear à l'ordre 1, est équivalent à un Euler implicite.

En réinjectant dans f : $f(x_{n+1}, \frac{x_{n+1} - \psi}{h_p \beta}, p, t) = 0$ (3.30)

Connaissant le point (t_n, x_n, \dot{x}_n) , le prédicteur (3.29) permet de déterminer une valeur initiale pour x_{n+1} . Cette valeur initiale sert pour la résolution du système d'équations algébriques non linéaires (3.30) par une méthode de type Newton Raphson :

$$x_{n+1}^k = x_{n+1}^{k-1} + \Delta x^{k-1} \quad (3.31)$$

Avec $m \Delta x^{k-1} = -f$ où m est l'opérateur dynamique (3.32)

$$m = \frac{\partial f}{\partial x} = \frac{\partial f}{\partial x} + \frac{1}{h_p \beta} \frac{\partial f}{\partial \dot{x}} \quad (3.33)$$

Remarque : ce qui explique que pour résoudre le système EDA par la méthode Gear, il faut

connaître $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial \dot{x}}$, comme énoncé au paragraphe III.1.

Lors de la résolution du système EDA, la matrice du correcteur m est factorisée. Cependant, afin de réduire le temps calcul, la matrice Jacobienne n'est pas recalculée systématiquement à chaque itération mais elle est mise à jour quand c'est nécessaire, car cette mise à jour et la

factorisation sont des étapes très coûteuses en temps calcul. Au final, dans le correcteur, le Jacobien est factorisé et mis à jour avec des fréquences variables.

Trois méthodes principales émergent pour résoudre le système EDA global : combinaison de l'EDA initial et des sensibilités. Les articles de Li et al (2000) et Feehery et al (1997) comparent ces méthodes avec des critères comme le temps calcul, l'efficacité.... Ces trois méthodes sont détaillées dans les paragraphes qui suivent.

Méthode directe décalée (staggered direct method)

Traditionnellement, le système EDA et celui des sensibilités sont résolus en séquence, la résolution tirant partie de la linéarité des équations de sensibilité. Cette méthode est nommée : *Méthode directe décalée*.

Cette méthode se base sur le constat que les équations différentielles pour les variables d'état sont indépendantes des variables de sensibilité. Dans un premier temps, le système EDA initial est résolu sur chaque pas. Après la convergence du correcteur pour les variables d'état, les matrices contenant les termes des équations de sensibilité sont obtenues au pas de temps courant. L'obtention exacte de ces termes, nécessite la mise à jour de la matrice Jacobienne à chaque pas de temps. Bien que la matrice m (équation 3.32) soit basée sur les mêmes informations que la matrice du correcteur pour le système initial, il est nécessaire de la mettre à jour et de la factoriser à chaque pas d'intégration, afin d'obtenir et résoudre les équations de sensibilité exactes. Or, comme on l'a spécifié en amont, la matrice Jacobienne pour le correcteur du système initial est mise à jour que lorsque cela est utile, dans le but de diminuer le temps calcul. Par conséquent ce travail obligatoire à chaque pas de temps est très consommateur en temps calcul.

Les équations de sensibilité sont discrétisées avec un schéma numérique identique aux variables d'état :

$$\dot{s}_{n+1} = \frac{s_{n+1} - \psi}{h_p \beta} \tag{3.34}$$

Dans ces conditions, les équations de sensibilité deviennent :

$$\frac{\partial f}{\partial \dot{x}_{n+1}} \left(\frac{s_{i_{n+1}} - \psi}{h_p \beta} \right) + \frac{\partial f}{\partial x_{n+1}} s_{i_{n+1}} + \frac{\partial f}{\partial p_i} = 0 \text{ soit :} \tag{3.35}$$

$$\underbrace{\left(\frac{1}{h_p \beta} \frac{\partial f}{\partial \dot{x}_{n+1}} + \frac{\partial f}{\partial x_{n+1}} \right)}_m s_{i_{n+1}} = \frac{\psi}{h_p \beta} \frac{\partial f}{\partial \dot{x}_{n+1}} - \frac{\partial f}{\partial p_i} \tag{3.36}$$

Après cette discrétisation, l'équation précédente est linéaire, et peut être résolue directement pour obtenir $s_{i_{n+1}}$ (sans itération). La résolution de ce système linéaire nécessite le calcul de la matrice Jacobienne à chaque pas de temps et le stockage auxiliaire de la matrice $\frac{\partial f}{\partial x_{n+1}}$ (2nd membre). Dans certains cas, la matrice Jacobienne est mal conditionnée, ce qui peut résulter en un échec de la méthode (exemple dans Li et al (2000)), ceci à cause du 2nd membre de l'équation 3.36 qui peut être très grand et peut introduire une erreur d'arrondi importante. Cette importante erreur d'arrondi peut causer un arrêt de la méthode lors des tests d'erreur. Lorsque ce cas se présente Li et al (2000) propose une modification de l'équation 3.36.

Cette méthode est parfois considérée comme inefficace (en temps calcul) car elle nécessite d'évaluer et de factoriser le Jacobien à chaque pas, ce qui est souvent la partie la plus coûteuse de la résolution. Toutefois, pour certains problèmes particuliers, cet inconvénient disparaît comme par exemple lorsque le coût d'évaluation d'une fonction est plus important que celui de la factorisation de la matrice Jacobienne (problème avec des cinétiques chimiques Li et al (2000)).

Méthode de correction simultanée (simultaneous corrector method)

Le constat de départ de cette méthode est que l'ajout des équations de sensibilité au système initial ne change pas le type de problème : système EDA. Cette méthode résout le système précédent en tant qu'un système EDA global, sans prendre en considération la linéarité des équations de sensibilité, et le fait que ces équations de sensibilité peuvent être connues exactement après avoir résolu le système initial.

Le système $F(X, \dot{X}, t, p) = 0$ (système initial + équations de sensibilités) est à résoudre avec $X(x, s_1, \dots, s_{np})$ comme variables.

La stratégie de résolution reste bien évidemment la même que celle présentée pour le système EDA initial, par contre l'opérateur dynamique devient :

$$M = \begin{pmatrix} m & & & & \\ m_1 & m & & & \\ \cdot & & \cdot & & \\ \cdot & & & \cdot & \\ m_p & & & & m \end{pmatrix} \text{ avec } m_i = \frac{\partial m}{\partial x} s_i + \frac{\partial m}{\partial p_i} \quad (3.37)$$

Par défaut, dans les intégrateurs mettant en œuvre cette méthode, le Jacobien M n'est pas calculé. En contrepartie, il est approximé par son block diagonal dans les itérations du Newton. Maly et Petzold (1996) montrent que sans cette approximation, le Newton a une

convergence quadratique par contre la convergence devient normale avec le Jacobien modifié. De plus, cette approximation a de grande chance d'augmenter le nombre d'itérations nécessaires dans le Newton.

Dans la méthode de correction simultanée, la matrice Jacobienne m est évaluée et factorisée seulement lorsqu'avec la version en cours de résolution, l'itération du Newton ne converge pas (ou est susceptible de ne pas converger).

Cette méthode améliore la méthode précédente en temps calcul car il n'y a plus besoin de factorisation additionnelle de matrices pour résoudre le système des équations de sensibilités.

Par contre son principal désavantage, est que les termes $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial \dot{x}}$, doivent être recalculés à

chaque itération pour générer la partie correspondant aux résidus des sensibilités. Bien que ce coût soit mineur, par rapport à la factorisation des matrices, dans certains cas il peut devenir significatif pour des problèmes de grande taille.

Méthode de correction décalée (staggered corrector method)

Dans son principe général, cette méthode est similaire à la méthode directe décalée. La différence principale est qu'à la place d'utiliser directement la factorisation de la matrice Jacobienne au pas courant pour résoudre les équations linéaires de sensibilités, elle prend la dernière factorisation de cette matrice Jacobienne à un des pas précédents, et résout le système linéaire des sensibilités par des itérations du type Newton. Par conséquent, elle évite des factorisations de la matrice Jacobienne ce qui permet de réduire le temps calcul surtout pour des problèmes de grande taille.

Après convergence du correcteur du système EDA initial, le système des équations de sensibilités devient :

$$A \begin{bmatrix} \dot{s}_i \\ s_i \end{bmatrix} = -\frac{\partial f}{\partial p_i} \quad (3.38) \text{ avec } A = \begin{bmatrix} \frac{\partial f}{\partial \dot{x}} \\ \frac{\partial f}{\partial x} \end{bmatrix} \quad (3.39)$$

Avec comme objectif d'éviter la factorisation de A (pour la résolution du système linéaire), 3.38 peut être résolue en utilisant la formule itérative suivante :

$$s_{i_{n+1}}^{k+1} = s_{i_{n+1}}^k - J^{-1} \left(m s_{i_{n+1}}^k - \frac{\psi}{h_p \beta} \frac{\partial f}{\partial \dot{x}_{n+1}} + \frac{\partial f}{\partial p_i} \right) \quad (3.40)$$

Avec J^{-1} *, la matrice jacobienne factorisée (mise à jour ou non) qui est utilisée (et sauvegardée) lors des itérations du correcteur pour les variables d'état. A noter que m représente le Jacobien au pas courant mais non factorisé. m et $\frac{\partial f}{\partial p_i}$ sont calculés une fois par pas de l'intégrateur : après convergence du correcteur pour le système EDA initial et avant le correcteur sur les sensibilités car ils dépendent uniquement d'informations provenant du système EDA.

Un des avantages de cette méthode est qu'il n'est pas nécessaire d'évaluer les équations de sensibilité pendant la résolution des variables d'état. Ceci peut réduire le temps calcul si les variables d'état demandent plus d'itérations que les variables de sensibilité.

Remarque : De plus, les itérations du correcteur pour les sensibilités ne demandent pas de factorisation supplémentaire de la matrice J^{-1} * car elle a été déjà réalisée dans le correcteur des variables d'état. C'est principalement cette raison qui rend cette méthode attractive, par rapport à d'autres.

Conclusion

Tirer des conclusions claires et précises sur une comparaison de ces méthodes est très difficile voire impossible. En effet, comme le démontre les tentatives de Li et al (2000) et Feehery et al(1997) de nombreux facteurs peuvent influencer la solution : type de problème, nombre de paramètres... A titre d'exemple le coût global en temps calcul dépend de ce qui se déroule durant un pas d'intégration (factorisation des matrices, mises à jour), du nombre total de pas sur l'horizon de temps mais également du choix de la méthode de calcul des résidus des équations de sensibilité et bien d'autres facteurs supplémentaires. A titre indicatif, le choix de la séquence de pas d'intégration retenue par l'intégrateur dépend de la méthode retenue pour intégrer les sensibilités. Cela est dû au fait que le choix du pas dépend des erreurs du Newton, or ces erreurs sont différentes d'une méthode à l'autre car les systèmes non linéaires résolus sont différents.

Concernant les méthodes de calcul des résidus, Feehery et al(1997) démontrent que pour le même problème, la méthode de correction décalée est moins coûteuse en temps calcul que la méthode de correction simultanée lorsque les résidus sont évalués par différences finies. Toutefois, peu de différences sont notables entre ces deux méthodes lorsque ces mêmes résidus sont calculés avec des dérivées directionnelles.

Le coût (en temps calcul) d'intégration du système initial est dominé par le coût de factorisation de la matrice, qui a lieu en moyenne tous les 5 à 10 pas (tout dépend du système). Pour les grands systèmes à structure creuse, le coût de la factorisation peut atteindre approximativement 85% du temps total d'obtention de la solution. Or pour la méthode directe

décagée, la factorisation est nécessaire à chaque pas de temps ce qui la rend moins efficace que les deux autres.

Cependant, afin de comparer les méthodes sur le coût généré par la factorisation et la mise à jour des matrices, lors du passage dans le correcteur des trois méthodes, les deux articles précités réalisent une comparaison sur un seul pas d'intégration. La conclusion de ces articles est la même et montre qu'en règle générale la méthode 3 (correction décalée) est plus efficace mais la méthode 2 (correction simultanée) a des performances peu éloignées (pour une analyse plus poussée de la comparaison se conférer aux deux articles). Toutefois, c'est la méthode 2 qui a été implantée dans la plateforme, en raison de sa simplicité et sa rapidité de mise en œuvre.

III.3 Paramétrisation du contrôle

Dans cette section, la structure et l'implémentation de la méthode sont décrites plus en détails, avec dans un premier temps une description de la paramétrisation du vecteur des fonctions de contrôle. Ensuite, les contraintes pour l'optimisation seront détaillées. Hormis, les contraintes provenant de la physique du modèle, la méthode par elle-même génère des contraintes additionnelles dues à la continuité des variables d'état et des fonctions de contrôles entre les intervalles d'intégration. Cette partie permet de démontrer comment ces contraintes seront gérées dans le code.

Sur chaque intervalle, chaque élément u_j ($j \in [1, Nu]$) du vecteur des fonctions de contrôle est représenté par des morceaux de polynômes d'ordre No . Chaque sous intervalle d'intégration est alors subdivisé en Nc intervalles de contrôle, figure 3.2.

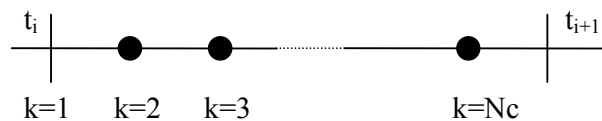


Figure 3.2 : Découpage d'un sous intervalle de temps pour la paramétrisation

La possibilité d'avoir des morceaux de polynômes sur un sous intervalle permet de pouvoir suivre plus finement la trajectoire optimale si le problème le nécessite. En contrepartie, cette option a pour conséquence d'augmenter fortement la taille du problème d'optimisation. En effet, sur chaque sous intervalle de contrôle d'un sous intervalle de temps, chaque fonction de contrôle possède un polynôme d'ordre No et donc $No+1$ coefficients à déterminer, soit au total $Nc \cdot (No+1)$ coefficients pour chaque élément u_j . Si $Nc=1$ alors le sous intervalle n'est pas subdivisé, par conséquent chaque u_j est représenté par un seul polynôme et donc un seul jeu de coefficients. Cette configuration correspond à la taille minimale du problème d'optimisation. Elle est très souvent suffisante pour les problèmes traités.

Lors de l'utilisation de cette option, il faut paramétrer l'ensemble des morceaux de polynômes représentant chaque u_j avec un minimum de paramètres. Ceci dans le but de ne pas trop augmenter la taille du système à optimiser, tout en assurant les continuités entre les morceaux de polynômes ainsi que celles de leurs dérivées. Ainsi chaque u_j est supposé de classe C^1 . Sur un intervalle de temps $[t_i ; t_{i+1}]$ quelconque, si N_c est le nombre de subdivisions, alors le temps d'une subdivision de contrôle est : $\Delta t_i = \frac{t_{i+1} - t_i}{N_c}$. Tout temps t sur ce sous intervalle appartient

à une subdivision de contrôle de sorte que :

$$k\Delta t_i \leq t - t_i < (k+1)\Delta t_i \quad (3.41)$$

Sur l'intervalle $[t_i ; t_{i+1}]$ l'élément u_j du vecteur de polynômes, à pour ordre N_o et s'écrit de la manière suivante sur la subdivision $[k ; k+1]$:

$$u_j(t) = u_{j,i}^{k+1,0} + u_{j,i}^{k+1,1} t^* + \sum_{q=2}^{N_o} u_{j,i}^{k+1,q} t^{*q} \quad (3.42)$$

Où : t^* est le temps adimensionnel $t^* = \frac{t - t_i - k\Delta t_i}{\Delta t_i} \in [0;1]$ (3.43)

- u : est un des coefficients du polynôme
- j : indice faisant référence à l'élément u_j du vecteur des polynômes
- i : indice faisant référence au sous intervalle d'intégration $[t_i ; t_{i+1}]$
- k : exposant faisant référence à la subdivision sur l'intervalle $[t_i ; t_{i+1}]$
- q : ordre du monôme

Ainsi la dérivée de (3.42) s'écrit :

$$\Delta t_i \dot{u}_j(t) = u_{j,i}^{k+1,1} + \sum_{q=2}^{N_o} q u_{j,i}^{k+1,q} t^{*q-1} \quad (3.44)$$

L'hypothèse que u_j soit C^1 sur le sous intervalle $[t_i ; t_{i+1}]$ impose des relations de continuité entre deux subdivisions successives, au point $t=(k+1) \Delta t_i + t_i$ par exemple :

$$\text{Continuité sur } u_j : u_{j,i}^{k+1,0} = u_{j,i}^{k,0} + u_{j,i}^{k,1} + \sum_{q=2}^{N_o} u_{j,i}^{k,q} \quad (3.45)$$

$$\text{Continuité sur } \dot{u}_j : u_{j,i}^{k+1,1} = u_{j,i}^{k,1} + u_{j,i}^{k,1} + \sum_{q=2}^{N_o} q u_{j,i}^{k,q} \quad (3.46)$$

Par remplacement successif des termes $u_{j,i}^{k,1}$ dans l'équation de continuité sur \dot{u}_j :

$$u_{j,i}^{k+1,1} = u_{j,i}^{1,1} + u_{j,i}^{k,1} + \sum_{k1=1}^k \sum_{q=2}^{N_o} q u_{j,i}^{k1,q} \quad (3.47)$$

De la même façon, le remplacement successif du terme dans l'équation de continuité sur u_j et en introduisant la formule précédente (3.47) on obtient :

$$u_{j,i}^{k+1,0} = u_{j,i}^{1,0} + ku_{j,i}^{1,1} + \sum_{k=1}^k \sum_{q=2}^{No} (q(k-1) + 1)u_{j,i}^{k,q} \quad (3.48)$$

Sans tenir compte des équations de continuité, précédentes (3.47) et (3.48), sur le sous intervalle $[t_i; t_{i+1}]$, il faut déterminer $Nc*(No+1)$ coefficients des morceaux de polynômes, pour connaître u_j . Par contre chaque équation de continuité donne $Nc-1$ relations entre ces coefficients, réduisant ainsi la détermination de u_j à la recherche de $Nc*(No-1)+2$ coefficients : $u_{j,i}^{1,0}; u_{j,i}^{1,1} u_{j,i}^{k,q}$ avec $k=1, \dots, Nc$ et $q=2, \dots, No$.

Au final sur l'ensemble de l'horizon de temps $[t_0; t_f]$, le nombre total de coefficients à déterminer est de $Nm*Nu*(Nc*(No-1)+2)$. Avec cette dernière relation, il est clairement visible que le nombre de coefficients croit rapidement avec le nombre de subdivisions de contrôle. Surtout que si l'option de subdiviser un sous intervalle d'intégration est retenue, alors tous les sous intervalles le seront.

Par rapport à une paramétrisation classique, l'approche présentée possède deux avantages majeurs issus de la continuité forcée des polynômes de contrôle et leurs dérivées. Le premier est directement relié au fait qu'un nombre plus réduit de coefficients est nécessaire $Nm*Nu*(Nc*(No-1)+2)$ au lieu de $Nm*Nu*Nc*(No+1)$ ce qui se traduit par moins d'équations de sensibilité à résoudre. Sans cette continuité C^1 , une discontinuité entre les morceaux de polynômes apparaîtrait dans les sensibilités par rapport aux paramètres $u_{j,i}^{k,0}$ et donc une initialisation cohérente serait nécessaire au début de chaque subdivision. Avec cette approche, l'intégration peut se dérouler sur l'ensemble du sous intervalle $[t_i; t_{i+1}]$ sans avoir besoin de la réinitialiser à chaque début de subdivision.

III.4 Les contraintes

La méthode présentée en amont impose une continuité pour les variables d'état et pour le vecteur des paramètres de contrôle aux points de découpage des intervalles d'intégration. Une condition de continuité simple (C^0) et une condition (C^1) pour les fonctions de contrôle sont imposées, ce qui résulte respectivement en des contraintes non linéaires pour les variables d'état et linéaires pour les fonctions de contrôle. Au final l'ensemble complet des contraintes pour le problème d'optimisation comprend les contraintes :

- du modèle, définies par l'utilisateur
- de continuité sur les variables d'état
- de continuité sur les fonctions de contrôle
- additionnelles (éventuelles) sur les fonctions de contrôle.

Pour ce qui est des contraintes définies par l'utilisateur lors de la modélisation, le traitement est classique, il n'est pas nécessaire de détailler ce point. Les autres contraintes découlent directement de la méthode retenue, il s'avère intéressant de les approfondir.

Continuité sur les variables d'état

Comme il a été précisé lors de la présentation générale de la méthode, les équations sont résolues sur des sous intervalles $[t_i ; t_{i+1}]$. La solution de l'intégration au temps t avec comme valeur initiale X_i au temps t_i se note $x(t, t_i, X_i, p, u)$. Aux bornes des sous intervalles, les conditions de continuité sont assurées par des contraintes non linéaires :

$$C_i : X_{i+1} - x(t_{i+1}, t_i, X_i, p, u) = 0, C_i \text{ est de taille } N_{x+1}. \quad (3.49)$$

Remarque : C_i est de dimension N_{x+1} en incluant une variable supplémentaire pour la fonction coût.

Soit $X = (X_2, X_3, \dots, X_{Nm+1}) \in \mathbb{R}^{Nm*(N_{x+1})}$ et $u = (u_1, u_2, \dots, u_{Nm}) \in \mathbb{R}^{Nm*Nu*(2+Nc*(No-1))}$

X_1 n'est pas dans le vecteur X car il correspond à l'état initial pour $t=t_1$.

Le système formé par ces contraintes s'écrit alors :

$$C(X, p, u) = \begin{bmatrix} C_1 \\ C_2 \\ \cdot \\ \cdot \\ C_{Nm} \end{bmatrix} = \begin{bmatrix} X_2 - x(t_2, t_1, X_1, p, u_1) \\ X_3 - x(t_3, t_2, X_2, p, u_2) \\ \cdot \\ \cdot \\ X_{Nm+1} - x(t_{Nm+1}, t_{Nm}, X_{Nm}, p, u_{Nm}) \end{bmatrix} \quad (3.50)$$

Le Jacobien de ces contraintes par rapport aux variables d'optimisation peut s'écrire :

$$J = [J_p ; J_X ; J_u]$$

Où

$$J_p = \begin{bmatrix} -\frac{\partial x(t_2)}{\partial p} \\ -\frac{\partial x(t_3)}{\partial p} \\ \cdot \\ \cdot \\ -\frac{\partial x(t_{Nm+1})}{\partial p} \end{bmatrix} \text{ vecteur de dimension } Nm*(N_{x+1}) \quad (3.51)$$

$$J_X = \begin{bmatrix} I & & & & \\ -\frac{\partial x(t_3)}{\partial X_2} & I & & & \\ & -\frac{\partial x(t_4)}{\partial X_3} & I & & \\ & & & \ddots & \\ & & & -\frac{\partial x(t_{Nm+1})}{\partial X_{Nm}} & I \end{bmatrix} \quad \text{dimension [Nm*(Nx+1);Nm*(Nx+1)]} \quad (3.52)$$

$$J_u = \begin{bmatrix} -\frac{\partial x(t_2)}{\partial u_1} & & & & \\ & -\frac{\partial x(t_3)}{\partial u_2} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & -\frac{\partial x(t_{Nm+1})}{\partial u_{Nm}} \end{bmatrix} \quad \text{dimension [Nm*(Nx+1);Nm*(2+Nc*(No-1))]} \quad (3.53)$$

Une des grandes difficultés numérique de cette méthode résulte dans le calcul de la matrice J_X . Lorsque le nombre de variables N_x devient grand, le temps calcul pour calculer ce Jacobien devient prohibitif. Serban et Petzold (1996) proposent une alternative pour diminuer le temps calcul en modifiant les contraintes et le Jacobien. En contre partie, cette modification génère une modification de l'algorithme d'optimisation. Cette modification n'est pas mise en œuvre ici.

Continuité sur les fonctions de contrôle

Les conditions de continuité sur u_j et \dot{u}_j ($j=1, \dots, N_u$) au point t_i ($i=2, \dots, N_m$) conduisent à $2*(N_m-1)*N_u$ contraintes linéaires sur les paramètres de contrôle.

$$C_{j,i}^1 = u_{j,i}^{1,0} + N_c u_{j,i}^{1,1} + \sum_{k=1}^{N_c} \sum_{q=2}^{N_o} [q(N_c - k) + 1] u_{j,i}^{k,q} - u_{j,i+1}^{1,0} \quad (3.53)$$

$$C_{j,i}^2 = u_{j,i}^{1,1} + \sum_{k=1}^{N_c} \sum_{q=2}^{N_o} q u_{j,i}^{k,q} - \frac{\Delta t_{i+1}}{\Delta t_i} u_{j,i+1}^{1,1} \quad i=1, 2, \dots, N_m \quad (3.54)$$

Le Jacobien de ces contraintes donne :

$$\frac{\partial C_{j,i}^1}{\partial u_{j,i}^{1,0}} = 1 ; \quad \frac{\partial C_{j,i}^1}{\partial u_{j,i}^{1,1}} = N_c ; \quad \frac{\partial C_{j,i}^1}{\partial u_{j,i}^{k,q}} = q(N_c - k) + 1 ; \quad \frac{\partial C_{j,i}^1}{\partial u_{j,i+1}^{1,0}} = -1 \quad (3.55)$$

$$\frac{\partial C_{j,i}^2}{\partial u_{j,i}^{1,1}} = 1 ; \quad \frac{\partial C_{j,i}^2}{\partial u_{j,i}^{k,q}} = q ; \quad \frac{\partial C_{j,i}^2}{\partial u_{j,i+1}^{1,1}} = \frac{\Delta t_{i+1}}{\Delta t_i}$$

Avec $i=1, 2, \dots, N_m$

$$\begin{aligned} j &= 1, 2, \dots, Nu \\ q &= 2, 3, \dots, No \\ k &= 1, 2, \dots, Nc \end{aligned}$$

Les contraintes additionnelles

A cause du schéma de paramétrisation des fonctions de contrôle, si l'utilisateur définit des valeurs limites sur ces fonctions de contrôle (et les dérivées) alors elles ne peuvent être uniquement appliquées qu'au début de chaque intervalle d'intégration. Or ces limites sont valables sur tout l'intervalle d'intégration. Par conséquent pour obliger le respect de ces limites sur la globalité de l'intervalle, pour chaque vecteur de contrôle, $2*Nc$ contraintes linéaires sont ajoutées sur chaque intervalle i . Soit LS_{uj} et LI_{uj} les limites supérieures et inférieures pour la fonction de contrôle j et LS_{duj} et LI_{duj} ces mêmes limites pour la dérivée correspondant. Alors on impose :

$$\begin{aligned} LI_{uj} &\leq u_{j,i}^{k,0} + u_{j,i}^{k,1} + \sum_{q=2}^{No} u_{j,i}^{k,q} \leq LS_{uj} \\ \Delta t_i LI_{u'j} &\leq u_{j,i}^{k,1} + \sum_{q=2}^{No} qu_{j,i}^{k,q} \leq \Delta t_i LS_{u'j} \end{aligned} \tag{3.56}$$

L'ensemble complet des contraintes pour le problème discrétisé est la concaténation des : contraintes définies par l'utilisateur, contraintes de continuité sur les variables d'état, contraintes de continuité sur les fonctions de contrôle et les contraintes additionnelles sur le contrôle.

IV Conclusion

Ce chapitre a permis de mettre en évidence la méthode retenue pour la résolution du problème d'optimisation de systèmes dynamiques. La méthode à tir multiple possède de nombreux avantages par rapport à celles présentées dans le chapitre précédent :

- Elle permet de tirer profit de l'intégrateur de systèmes EDA DISCo, disponible dans la plateforme et par voie de conséquence des avantages que peut procurer cet intégrateur : détection d'événement, réduction d'index... Car dans le cas de méthodes simultanées, le système EDA n'est pas intégré (les équations différentielles sont discrétisées)
- Des problèmes complexes sont abordables par cette méthode. A cause de problème de programmation non linéaire de grande taille, au vu de la littérature, il apparait que les problèmes de contrôle optimal de procédés traités selon une approche simultanée, font intervenir des modèles de complexité plus modeste.
- La discrétisation de l'horizon de temps en sous intervalle, autorise la prise en compte directe de contraintes sur les variables d'état, ce que ne permet pas les méthodes séquentielles (mais possibilité avec la pénalisation de la fonction objectif).

- Un autre avantage issu du découpage de l'intervalle de temps est la parallélisation possible de l'intégration du système EDA sur les sous intervalles.

Comme toute méthode numérique elle a aussi ses inconvénients :

- Bien qu'elle soit possible, la prise en compte des contraintes sur les fonctions d'état peut s'avérer insuffisante dans certains cas, du fait de leur application seulement aux points limites des sous intervalles. Le chapitre suivant mettra en évidence cette limitation.
- Comme il a été précisé plus haut, une des grandes difficultés numériques de cette méthode résulte dans le calcul de la matrice J_X . Lorsque le nombre de variables N_x devient grand, le temps calcul pour estimer ce Jacobien devient prohibitif. L'amélioration de Serban et Petzold (1996) sera à mettre à œuvre.
- Et bien évidemment ce type de méthode ne permet pas de garantir la globalité de l'optimum.

Au final, la version disponible dans le noyau numérique de la plateforme Prodhys n'est que la première version de l'implantation de cette méthode. De nombreux points sont perfectibles afin d'apporter des évolutions à la version mise en œuvre. Après cette présentation de la méthode, le chapitre suivant permet de la tester sur des exemples d'application.

CHAPITRE 4

Illustration en Génie des Procédés

I Introduction

Ce chapitre a pour objet de mettre en pratique la méthode d'optimisation dynamique au travers d'exemples. Mais avant l'optimisation dynamique, l'utilisateur doit réaliser plusieurs étapes. La première concerne la modélisation du problème : écriture du modèle (avec toutes les difficultés que cette étape comporte), détermination de la fonction objectif ainsi que la détermination des contraintes d'optimisation (contraintes sur les variables d'état, sur le contrôle. Quant aux contraintes de continuité elles sont générées automatiquement). Toutefois, avant de passer à la phase d'optimisation dynamique, il est conseillé d'intégrer le système EDA seul puis avec les sensibilités afin de passer outre les premières difficultés concernant ces étapes. Cette étape préliminaire peut s'avérer riche en connaissances sur le modèle pour la phase suivante d'optimisation.

Dans le cadre de ce chapitre trois applications sont développées :

- La première concerne une application test, elle est centrée sur un réacteur fed batch dont l'objectif est de maximiser la production d'une protéine.
- La seconde, est également une application directe de la méthode au travers d'un exemple issu de la littérature concernant un réacteur batch. L'intérêt de cet exemple est qu'il permet de tester la méthode avec la présence d'une contrainte sur une variable d'état.

- Le troisième exemple n'est pas strictement une application directe de la méthode du chapitre 3, mais il montre plutôt l'étendue des fonctionnalités que peut réaliser la plate forme PrODHyS avec l'ajout de la partie optimisation. Il s'intéresse à un problème d'identification de paramètres cinétiques dans le cadre d'une réaction sur un lit fixe.

II Réacteur Fed Batch

Ce premier exemple fait office de test pour la méthode proposée dans le chapitre 3. Cependant, l'implémentation de la méthode a subi un pré test pour valider le calcul des sensibilités. L'exemple présenté dans le chapitre précédent au paragraphe III.1, a été simulé avec le calcul analytique et numérique des équations de sensibilités. Les résultats étant similaires avec les deux modes de calcul, il a été décidé de simuler un exemple plus conséquent qui prend en compte des contraintes, notamment sur la fonction de contrôle. Il est issu des travaux de Park et Ramirez (1988) sur la production d'une protéine dans un réacteur fed-batch. Le but de cette opération est de maximiser la production de protéines secrétées dans le réacteur, pour cela la variable d'action est le débit d'entrée u . A noter que cet exemple a également été traité par Luus et Rosen (1991) par une méthode de programmation dynamique itérative.

II.1 Le modèle

Dans cette partie le modèle est présenté sans détailler les conditions de son établissement (hypothèses etc...).

$$\begin{aligned} \frac{dx_1}{dt} &= \frac{4.75 A(x_2 - x_1)}{0.12 + A} - \frac{u}{x_5} x_1 \\ \frac{dx_2}{dt} &= \frac{x_3 x_4 \exp(-5x_4)}{0.1 + x_4} - \frac{u}{x_5} x_2 \\ \frac{dx_3}{dt} &= \left(A - \frac{u}{x_5} \right) x_3 \quad \text{avec } A = \frac{21.87x_4}{x_4^2 + 62.9x_4 + 25} \text{ et } x_0 = (0, 0, 1, 5, 1) \quad (4.0) \\ \frac{dx_4}{dt} &= -7.3Ax_3 - \frac{u}{x_5} (x_4 - 20) \\ \frac{dx_5}{dt} &= u \end{aligned}$$

x_1 : concentration de protéines secrétées

x_2 : concentration totale de protéines

x_3 : densité de cellules dans la culture

x_4 : concentration de glucose dans la culture

x_5 : volume de la culture

L'objectif de cette optimisation dynamique est de maximiser le produit désiré dans le réacteur. Ce qui se traduit par la fonction objectif suivante :

$$\text{Min } J = -x_1(t_f) * x_5(t_f) \quad (4.1)$$

Avec t_f fixé à 15 heures. La variable d'action est le débit d'alimentation qui est borné :

$$0 \leq u \leq 10 \quad (4.2)$$

II.2 Résultats

Les résultats trouvés avec les paramètres précédents sont comparés à ceux proposé par Luus et Rosen (1991). Toutefois ces derniers ont considéré la commande constante par morceau et l'intervalle de temps est subdivisé en 40 sous intervalles pour la méthode de programmation dynamique itérative. Afin de pouvoir comparer les résultats des deux approches, les conditions de simulation des auteurs sont reprise, à savoir : l'horizon de temps découpé en 40 sous intervalles et des polynômes constants sur ces sous intervalles. Les résultats sont présentés sur deux graphiques séparés pour une meilleure lisibilité car la différence entre les deux études est minime.

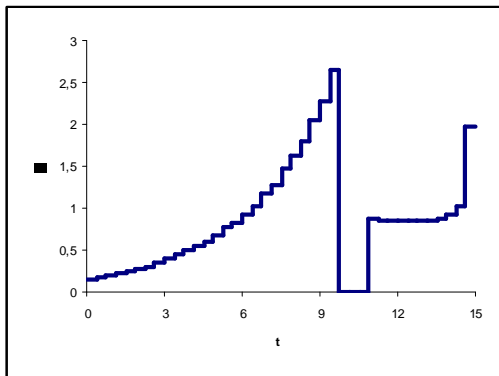


Figure 4.1 : Résultats de Rosen et Luus (1991)

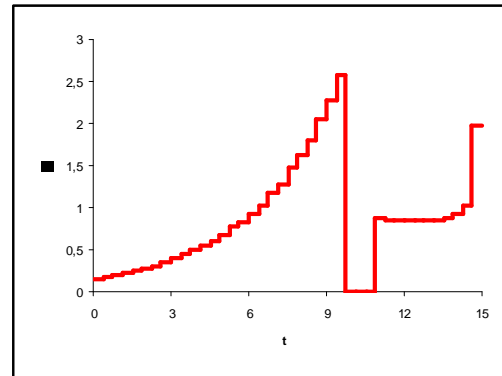


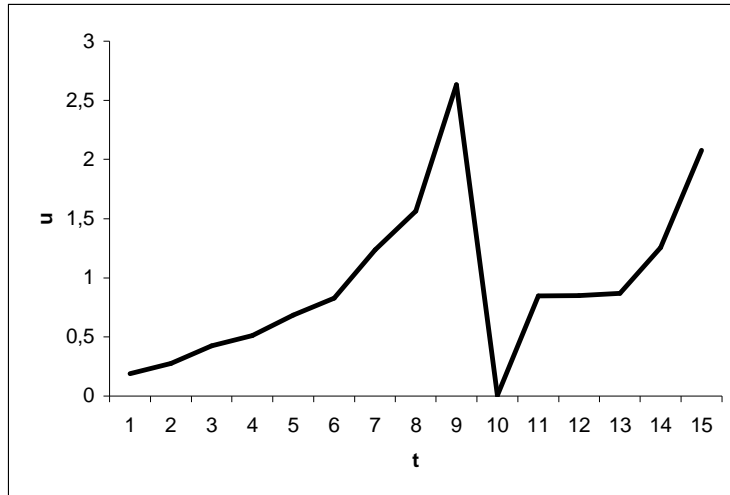
Figure 4.2 : Nos Résultats

Il est à noter que les courbes sont très similaires, de plus la valeur du critère est sensiblement la même dans les deux cas : -32.6 pour Luus et Rosen , et -32.3 ici. La seule petite différence notable provient du maximum pour la fonction de contrôle qui est un peu moins élevé dans nos résultats. Malheureusement, la comparaison entre les deux méthodes ne peut aller plus loin car les auteurs ne donnent pas plus de précision sur les performances de leur méthode.

Toutefois, sur cet exemple il nous a paru intéressant de faire varier quelques paramètres de la méthode. Dans un premier temps le nombre de subdivision de l'horizon de temps varie : 15, 20, 40. Ce découpage a très peu d'influence sur le critère (et sur le résultat) puisqu'il respectivement de : -31.7 pour N=15, -32.1 pour N=20 et -32.3 pour N=40. La seule différence notable entre ces différentes simulations provient du temps calcul qui est bien

évidemment accru lorsque le nombre de subdivision augmente. Ce temps calcul reste relativement faible vu que le système à résoudre est de petite taille. La figure 4.3 représente la commande pour $N=15$ et des polynômes linéaires sur les sous intervalles de temps.

Figure 4.3 : Résultats pour $N=15$

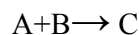


La résolution de ce problème a été abordée avec deux façons différentes de calculer les sensibilités. Tout d'abord ces équations de sensibilités sont calculées automatiquement par le programme via l'utilisation de l'approche par dérivées directionnelles (présentation au chapitre précédent). Cependant, le modèle de cet exemple est relativement simple et le calcul analytique des équations de sensibilités est accessible sans trop de difficulté. C'est pourquoi dans un second temps la simulation est reprise avec les mêmes paramètres pour la méthode, seule change la méthode de calcul des sensibilités. La différence notable entre ces deux méthodes se fait essentiellement sur le nombre d'itérations qui est plus conséquent dans le cas des dérivées directionnelles : passage de 26 itérations à 42. Cette différence se traduit bien évidemment sur un gain en temps calcul et elle s'explique par une meilleure précision sur les sensibilités et donc sur les gradients dans le cas du calcul analytique.

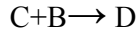
III Optimisation d'un réacteur batch

Le deuxième exemple est plus complexe car en plus de prendre en considération des contraintes sur les fonctions de contrôle, il y en a également sur les variables d'état du système. Cet exemple est issu des travaux de Vassiliadis et al (1994b), il permet d'illustrer les limites de la méthode implémentée.

Le problème abordé s'attache à résoudre un problème de contrôle optimal dans un réacteur batch où se déroule la réaction :



Avec comme réaction secondaire



Ces deux réactions sont fortement exothermiques, par conséquent le mélange direct des quantités nécessaires de A et B est à éviter. La politique retenue consiste à charger initialement le réacteur avec l'ensemble du produit A à traiter et de réaliser une coulée lente de B pur pendant la réaction. La chaleur dégagée par les réactions est évacuée à la paroi par une double enveloppe avec circulation d'eau. A des fins de comparaison avec les résultats des auteurs, l'exemple sera simulé dans les mêmes conditions : même modèle, conditions initiales, contraintes...

III.1 Modélisation

Avec les hypothèses simplificatrices, le comportement du système décrit ci-dessus est modélisé par le modèle mathématique suivant :

$$\begin{aligned}
 \dot{M}_A &= -Vr_1 \\
 \dot{M}_B &= F_B - V(r_1 + r_2) \\
 \dot{M}_C &= V(r_1 - r_2) \\
 \dot{M}_D &= Vr_2 \\
 \dot{H} &= F_B h_{BF} - Q - V(r_1 \Delta H_1 + r_2 \Delta H_2) \\
 V &= \frac{M_A}{\rho_A^0} + \frac{M_B}{\rho_B^0} + \frac{M_C}{\rho_C^0} + \frac{M_D}{\rho_D^0} \\
 M_i &= VC_i \quad i = A, B, C, D \\
 r_1 &= k_1 C_A C_B \\
 r_2 &= k_2 C_B C_C \\
 H &= M_A h_A(T) + M_B h_B(T) + M_C h_C(T) + M_D h_D(T) \\
 k_j &= A_j e^{-E_j/RT} \\
 M_T &= \sum_i M_i
 \end{aligned} \tag{4.3}$$

Avec : M_i : rétention molaire du constituant i

C_i : concentration molaire du constituant i

M_T : rétention molaire totale

H : enthalpie totale du mélange (énergie)

T : température

r_j : vitesse de la réaction j

V : volume liquide

- h_i : enthalpie du constituant i
- k_j : constante cinétique de la réaction j
- h_{BF} : enthalpie du courant d'entrée
- F_B : débit molaire de coulée
- Q : énergie évacuée à la paroi

Pour calculer les densités liquides et les enthalpies, le mélange est considéré comme idéal. Les données thermodynamiques et cinétiques sont résumées dans les tableaux 4.1 et 4.2.

Réaction	A_j ($m^3 \cdot mol^{-1} \cdot s^{-1}$)	E_j/R (K)	ΔH_j ($kJ \cdot mol^{-1}$)
1	0.008	3000	-100
2	0.002	2400	-75

Tableau 4.1 : Données Cinétiques

Composant	ρ_i^0 ($mol \cdot m^{-3}$)	α_i ($kJ \cdot mol^{-1} \cdot K^{-1}$)*	β_i ($kJ \cdot mol^{-1} \cdot K^{-2}$)*
A	11 250	0.1723	$0.474 \cdot 10^{-3}$
B	16 000	0.2000	$0.500 \cdot 10^{-3}$
C	10 400	0.1600	$0.550 \cdot 10^{-3}$
D	10 000	0.1550	$0.232 \cdot 10^{-3}$
$T_{ref} = 298$ K	$h_{BF} = 20$ $kJ \cdot mol^{-1}$		

* $h_i = \alpha_i(T - T_{ref}) + \frac{\beta_i}{2}(T^2 - T_{ref}^2)$

Tableau 4.2 : Données Thermodynamiques

Initialement le réacteur contient 9000 moles de A pur à une température de 350 K. Le but est de maximiser la production du composant C avec comme variable d'action : le débit d'alimentation et la quantité de chaleur évacuée à la paroi, respectivement F_B et Q . Cependant, des contraintes sont imposées sur ces variables, le débit ne peut pas dépasser 10 $mol \cdot s^{-1}$ et la puissance évacuée à la paroi doit être inférieure ou égale à 1000 kW. Pour éviter un refroidissement excessif du mélange en fin de réaction, la température finale est également contrainte : elle doit être comprise entre 295 K et 300 K. De plus la durée totale de la réaction est laissée libre, c'est une solution du problème d'optimisation :

$$\underset{t_f, F_B, Q}{Min} - M_c(t_f) \tag{4.4}$$

Soumis au système EDA et aux contraintes :

$$\begin{aligned}0 &\leq F_B \leq 10 \\0 &\leq Q \leq 1000 \\295 &\leq T(t_f) \leq 300\end{aligned}\tag{4.5}$$

III.2 Résultats

Dans la méthode de résolution, le choix du nombre d'intervalles de discrétisation, d'intervalles de contrôle et l'ordre des polynômes pour la représentation du contrôle est réalisé par l'utilisateur. Par exemple, la façon dont le contrôle est implémenté sur le procédé réel peut dicter la forme de la fonction de contrôle. Toujours à des fins de comparaison avec les résultats de Vassiliadis et al (1994b), les fonctions de contrôle sont considérées constantes par morceaux sur chaque sous intervalle de temps.

Avant de présenter les résultats, une précision est à apporter concernant le temps final non fixé. Dans ces conditions, il faut spécifier un paramètre p_{tf} qui correspond au temps final. Dans ce cas le modèle est réécrit avec le temps adimensionnel $\tau = t / p_{tf}$, par conséquent toutes les dérivées par rapport au temps sont transformées par :

$$\frac{d}{dt} = \frac{d}{d\tau} \frac{d\tau}{dt} = \frac{1}{p_{tf}} \frac{d}{d\tau}\tag{4.6}$$

Enfin l'intégration sera réalisée sur τ qui varie sur l'intervalle $[0 ; 1]$, et p_{tf} devient un paramètre déterminé lors de l'optimisation.

Le temps final est initialisé à 1000s, cet horizon de temps est découpé en 10 sous intervalles de 100s avec $F_B=5 \text{ mol.s}^{-1}$ et $Q=0$ constants pour tout t de l'horizon de temps.

La solution de ce problème donne un temps optimum de 2410s avec une production de 3782 moles de C. Ces résultats sont comparables à ceux des auteurs, il en est de même pour les résultats présentés sous forme graphique, figures 4.4, 4.5, 4.6, 4.7 (les résultats étant semblables seuls les nôtres sont représentés pour gagner en clarté). Cependant, une différence notable apparaît sur le temps final qui est de 2690s pour Vassiliadis et al (1994b), mais dans leur cas, la quantité de chaleur Q est constante aux environs de 900kW durant la phase de refroidissement tandis que pour nous elle est de 1000 kW sur cette période.

La politique optimum de coulée stipule que le constituant B est introduit durant approximativement, la première moitié du temps opératoire. Lorsque la coulée de B est stoppée pratiquement au même instant le système de refroidissement est mis en marche pour amener la température à la consigne imposée en fin de réaction. Sur la figure 4.6, il est clairement visible que la réaction est terminée dès que l'alimentation de B est arrêtée.

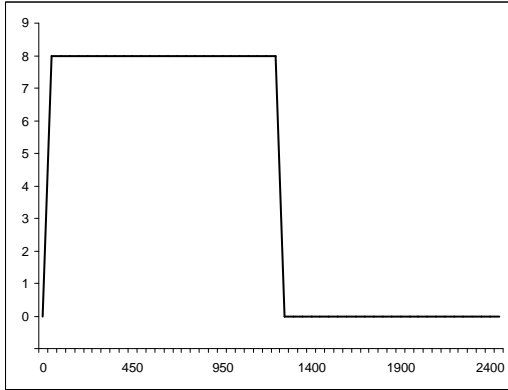


Figure 4.4 : Débit de coulée

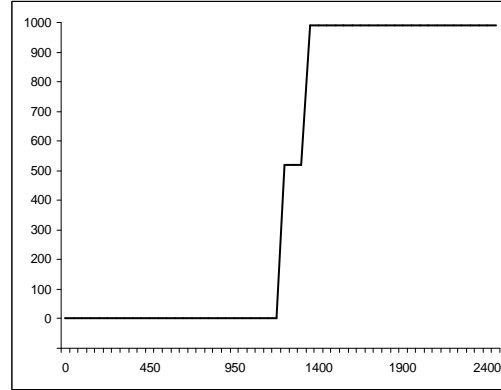


Figure 4.5 : Chaleur évacuée

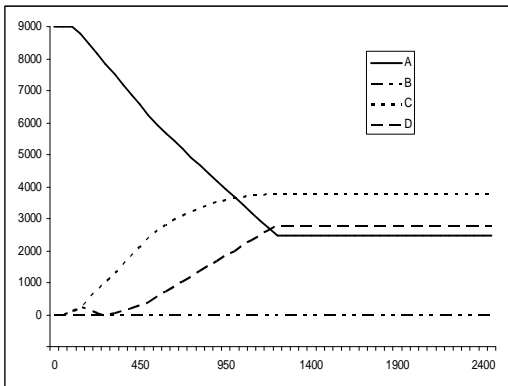


Figure 4.6 : Réention Molaire

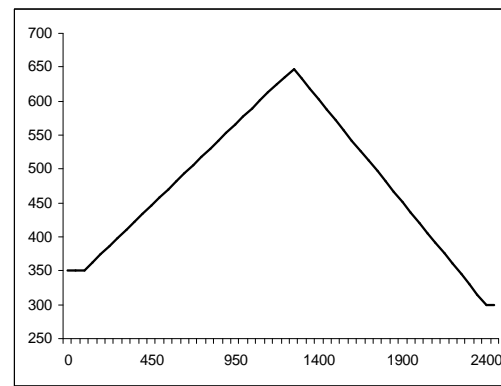


Figure 4.7 : Température

Maintenant, pour des raisons de sécurité, une contrainte supplémentaire est ajoutée au système précédent : la température du milieu réactionnel ne doit pas excéder 520 K durant toute la réaction. Sur les résultats précédents, il est clairement visible que cette contrainte n'est pas respectée. Il convient alors de l'imposer pour l'ensemble de l'horizon de temps.

III.3 Résolution avec contrainte de température

Le problème précédent est repris avec les mêmes conditions opératoires et initiales mais avec une contrainte supplémentaire :

$$T(t) \leq 520K \quad (4.7)$$

Le simple ajout de cette contrainte a une influence importante sur le résultat et le temps de résolution (Vassiliadis et al (1994b) font exactement le même constat malgré une méthode de résolution différente : ils utilisent une approche séquentielle couplée à l'ajout d'une variable différentielle avec une contrainte au temps final sur cette dernière pour prendre en compte la contrainte sur la température, méthode présentée au chapitre 2, paragraphe IV.1. La méthode présentée au chapitre 3 permet seulement d'imposer la contrainte de température au point de discrétisation de l'horizon de temps. Or si l'on repart avec la discrétisation initiale 10 intervalles de temps de 100s chacun, la contrainte est bien respectée au point de subdivision

(toutes les 100s). Mais entre ces derniers, ces contraintes ne sont pas directement appliquées aux variables d'état, autorisant ponctuellement le non respect de celles-ci : ce qui est le cas puisque la température dépasse les 520 K entre les bornes de certains intervalles.

Pour remédier à cet inconvénient deux alternatives sont possibles :

- Adoptée la même stratégie que Vassiliadis et al (1994b), avec l'ajout d'une variable supplémentaire et une contrainte sur cette dernière, ce qui au final donne la même méthode que les auteurs et des résultats semblables.
- Réduire la taille des intervalles de temps de façon à avoir une discrétisation plus fine et donc forcer 'indirectement' la température à respecter la contrainte.

La deuxième stratégie a été mise en œuvre avec des intervalles de temps de 30s. Par rapport au cas précédent le temps calcul a été quasiment doublé. Ce qui s'explique par le nombre plus élevé d'intervalle. Au final, les résultats obtenus sont représentés sur les figures 4.8, 4.9, 4.10, 4.11. La production de C est légèrement réduite à 3610 moles mais en un temps opératoire de 1953s. Ce temps opératoire est réduit notamment grâce à la diminution du temps de refroidissement du fait d'une température de départ du refroidissement plus faible. A noter que les résultats sont comparables à ceux des auteurs.

Remarque : Cette fois ci, les durées opératoires sont quasiment identiques 1936s pour les auteurs contre 1953s pour nous, car pendant la période de refroidissement la chaleur évacuée est bien de 1000 kW pour les deux études.

Sur les figures 4.8 , 4.9, on remarque qu'il existe une période où la coulée de B est active et le refroidissement aussi (contrairement au cas précédent). Cette période correspond à l'intervalle de temps où la contrainte sur la température est active.

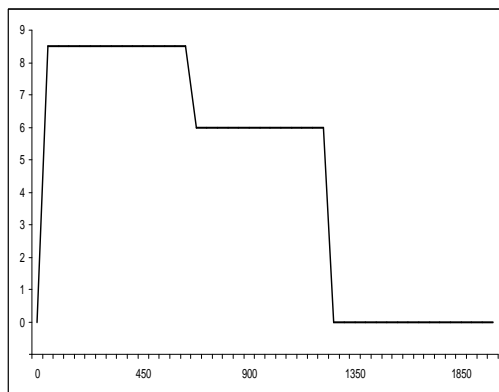


Figure 4.8 : Débit de coulée

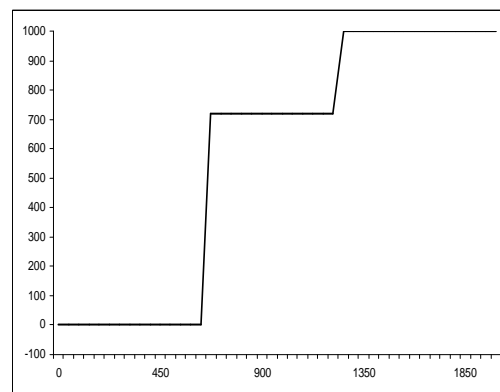


Figure 4.9 : Chaleur évacuée

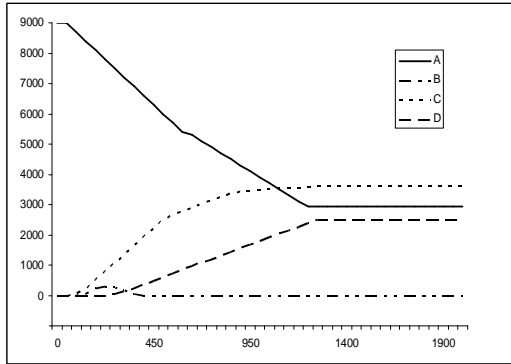


Figure 4.10 : Rétention Molaire

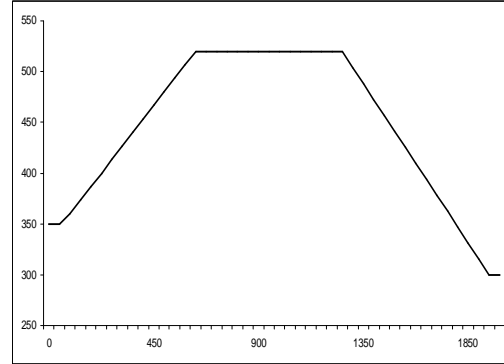


Figure 4.11 : Température

IV Identification Paramétrique

Cet exemple a servi d'application de la méthode dans le cadre d'une thèse concernant le projet ITER (International Thermonuclear Experimental Reactor). Compte tenu du sujet stratégique de cette thèse, il est bien évident que les résultats sont confidentiels. Par conséquent des zones d'ombre seront volontairement laissées dans la présentation du problème ainsi que certains graphiques illustrant les résultats seront dépourvus d'échelles. Malgré ces contraintes, la compréhension et la présentation de la problématique n'en sont pas affectées.

IV.1 Contexte de l'étude

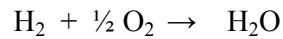
Présentation générale

Le but de cette étude était de maîtriser le risque hydrogène dans le cadre du projet ITER et de répertorier les solutions de mitigation qui ont été ou peuvent être envisagées pour répondre à cet objectif. Le réacteur expérimental ITER vise à démontrer la faisabilité de la maîtrise de la fusion thermonucléaire lors d'expériences dites de longues durées permettant de produire 400 à 5000 MW de puissance électrique sur des périodes de 400s. La fusion thermonucléaire est basée sur la réaction de deux isotopes de l'hydrogène : deutérium-tritium, productrice d'énergie. Les principes de sureté sur lesquels repose la conception de l'installation ITER ont pour objectif de protéger le personnel et l'environnement des risques potentiels liés à cette installation : risques toxiques, chimiques et radiologiques.

Le risque hydrogène

La présence d'hydrogène et de ses isotopes fait partie intégrante des risques chimiques, c'est pourquoi il a été étudié. Dans le réacteur de fusion ITER, plusieurs scénarios accidentels peuvent aboutir à la formation de quantités significatives d'hydrogène (toutefois, des systèmes passifs de sécurité sont mis en place permettant un déplacement de l'hydrogène et par conséquent du risque).

En cas de brèches, plusieurs réactions d'oxydation de composants par l'eau de refroidissement peuvent aboutir à la formation d'hydrogène et par conséquent à un risque de combustion. Outre ce problème, diverses sources sont possibles pour la présence d'hydrogène. Ce risque hydrogène provient de la réaction (fortement exothermique) entre l'hydrogène et l'oxygène, qui présente un fort potentiel explosif :

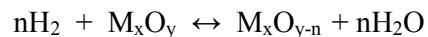


Cette réaction peut avoir lieu suivant deux modes : combustion lente du type propagation de flamme, ou au contraire suivant un mode explosif (déflagration, transition déflagration – détonation, détonation).

Il existe plusieurs moyens pour la mitigation du risque hydrogène :

- *Inertage par dilution* : injection de gaz inerte (hélium, azote...) pour se mettre dans des conditions opératoires où la réaction de combustion ne peut pas se dérouler.
- *Élimination de l'hydrogène* : deux techniques sont possibles ici : soit par la maîtrise de la réaction hydrogène/oxygène (igniteur catalytique), soit par l'élimination par voie chimique (en phase solide, liquide ou gaz).
- *Élimination de l'oxygène* : réaction avec des composés organiques (pyrogallol) ou par combustion du méthane.

Le but consiste à limiter la quantité d'hydrogène en cas de brèches. Compte tenu de contraintes supplémentaires (traitement d'un mélange hydrogène/vapeur d'eau, système passif...), le choix s'est orienté vers l'élimination chimique en phase solide et plus particulièrement vers la réduction des oxydes métalliques en présence d'un promoteur :



Où M est un métal. Des analyses poussées ont permis de retenir un couple oxyde métallique/promoteur.

Malgré quelques données disponibles dans la littérature pour la réaction retenue, aucune ne correspond aux conditions opératoires prévues. Par conséquent, une campagne d'essais sur une installation pilote a été réalisée : les résultats de cette étude en dynamique permettront de déterminer les paramètres cinétiques (entre autres) par identifications avec le modèle phénoménologique.

Campagne expérimentale

Les essais pour déterminer la cinétique de la réaction sont réalisés sein d'une installation qui permet de faire réagir un gaz avec un solide disposé dans un réacteur à lit fixe. Des systèmes d'analyses permettent d'étudier : la composition du gaz avant et après la réaction, mesurer et réguler la température, caractériser le solide. Les mesures sont réalisées pour une configuration et des conditions expérimentales dites de références. Toutefois, certains paramètres ou conditions sont modifiés pour certains essais. Ces paramètres ont été choisis en fonction de l'influence directe qu'ils peuvent avoir sur la cinétique mais aussi sur

l'écoulement : taille du lit (quantité de solide), taille des grains, température d'entrée du gaz, teneur en hydrogène du gaz entrant, débit d'entrée, teneur initiale du gaz en eau.

Afin de bien comprendre la physique du problème, les deux figures ci-dessous mettent en évidence les différents phénomènes présents lors des expériences :

Concernant les températures, dans un premier temps les températures évoluent rapidement sur l'ensemble du lit, ce qui met en évidence une réaction surfacique entre l'hydrogène et l'oxyde métallique. Cette évolution rapide est suivie de l'apparition d'un front réactionnel caractérisé par une forte augmentation de température à l'entrée du lit et par stagnation pour les autres points jusqu'à l'arrivée du front.

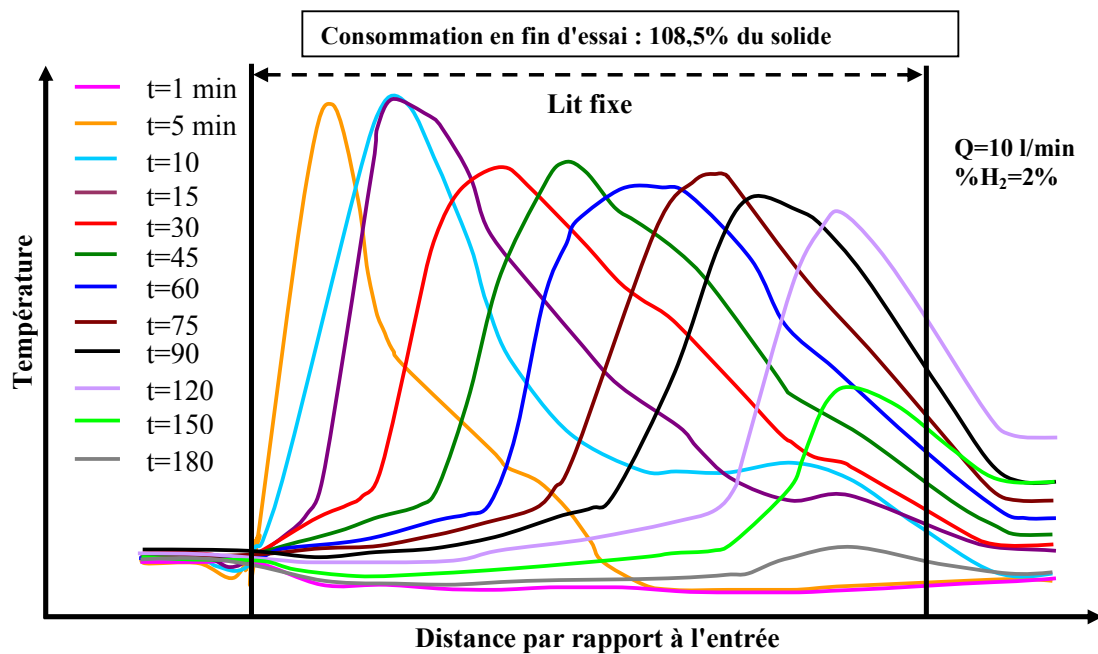


Figure 4.12 : Profil de température dans le lit fixe.

Sur la figure 4.13, concernant l'évolution de l'hydrogène, trois zones se distinguent correspondant à trois phénomènes différents :

- ◆ Zone 1 : le flux d'hydrogène en sortie de l'installation est pratiquement nul. La totalité de l'hydrogène introduit est éliminée. Cela correspond à la traversée du lit par le premier front de réaction (correspondant au premier front de température) qui serait essentiellement dû à la réaction à la surface des grains solides (cinétique chimique).
- ◆ Zone 2 : Augmentation progressive de la teneur en gaz en sortie du réacteur jusqu'à stabilisation. Durant cette période, l'efficacité d'élimination du procédé diminue progressivement. Sur cette zone, la réaction est limitée par la diffusion de l'hydrogène à travers la couche de cendres de l'oxyde métallique déjà formée. Le second front de

température caractérise cette étape qui se termine avec la consommation totale du catalyseur (théoriquement).

- ◆ Zone 3 : fin de la réaction avec un phénomène résiduel : la vitesse d'élimination de l'hydrogène est constante mais faible. Des hypothèses sont avancées pour expliquer ce phénomène mais il n'est pas clairement identifié, cependant il est consommateur d'hydrogène. De plus, il justifierait l'élimination de l'hydrogène, supérieure à la capacité théorique (100%) du solide dans le lit fixe.

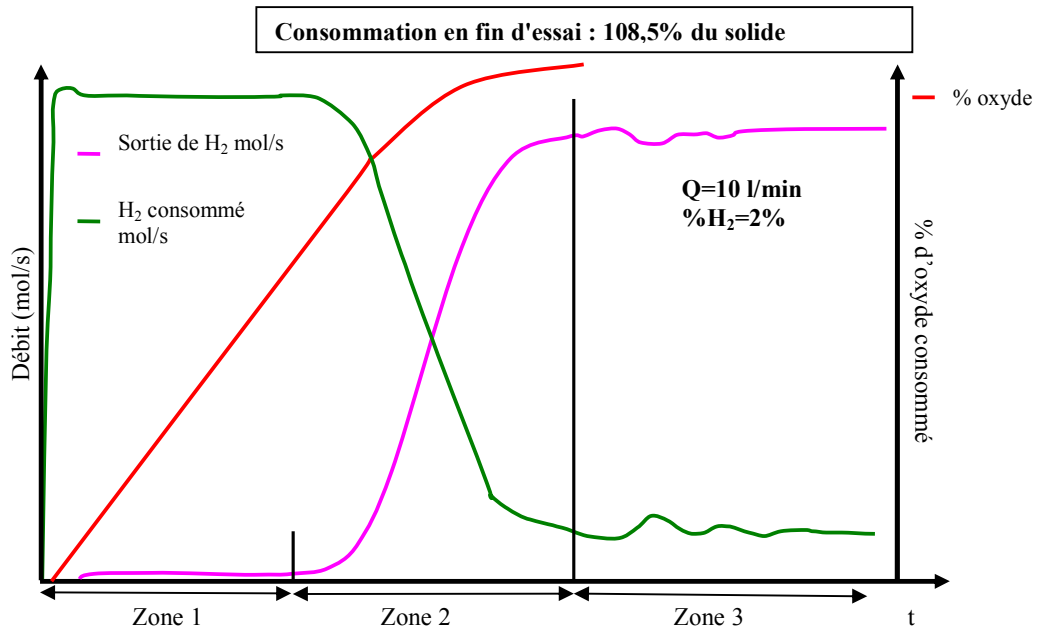


Figure 4.13 : Evolution des quantités d'hydrogène dans le gaz

La campagne expérimentale est bien plus riche en enseignement que les points abordés dans cette partie. Compte tenu de la confidentialité de l'exemple et de la finalité de ce manuscrit, il n'est pas obligatoire d'aller plus en profondeur dans la compréhension des phénomènes mis en jeu. Surtout qu'ils ne sont pas nuisibles à la compréhension de l'exemple. Cependant les éléments précisés ci-dessus éclairent les choix et les hypothèses de modélisation de ces études.

IV.2 Modélisation

Cette partie aborde la modélisation du système réactionnel étudié lors des essais. Elle détaille également, la mise en œuvre de l'identification paramétrique en adaptant la méthode présentée au chapitre 3. Au final, les résultats obtenus après simulation seront comparés aux résultats expérimentaux.

Modélisation de la réaction

De nombreux modèles de réaction gaz-solide ont été proposés dans la littérature, parmi les plus utilisés on retrouve : le modèle du grain à cœur rétrécissant et cendres adhérentes (et ses modèles dérivés), le modèle de catalyse modifié, le modèle de réaction homogène... Après analyse de ces modèles, couplée à la connaissance acquise sur le procédé durant la campagne expérimentale, le choix s'est naturellement orienté vers le modèle du grain à cœur rétrécissant et cendres adhérentes. Ce dernier considère que les grains de solide réagissent de la périphérie vers le centre du solide en formant une couche de cendres que le gaz doit traverser (Villiermaux (1993)). Les hypothèses d'un tel modèle sont :

- La réaction est supposée irréversible d'ordre 0 par rapport au solide.
- Les particules de solide sont des grains sphériques de structure uniforme (possibilité de prendre en considération des facteurs de forme).
- Les grains sont initialement non poreux, ils le deviennent avec l'apparition des cendres
- La densité du grain et celle du solide initial (pas de variation de dimension du grain en cours de réaction).
- Les grains sont supposés isothermes.
- La phase gaz est considérée comme idéale.

Ce modèle prend en considération de trois résistances au transfert :

- Le transfert du gaz à travers la couche limite externe
- La diffusion dans les pores de la couche de cendres
- La réaction chimique à l'interface solide non réagi/gaz

Ces trois résistances au transfert permettent de différencier trois régimes :

- *Régime de transfert dans le gaz* : la diffusion du réactif dans la couche limite externe est l'étape limitante
- *Régime de diffusion* : cette fois-ci, la diffusion du réactif dans les pores limite le transfert. Ce cas de figure ne peut avoir lieu qu'après la formation d'une couche de cendres d'épaisseur suffisante.
- *Régime chimique* : c'est la cinétique de la réaction qui limite le processus.

Dans la pratique c'est une combinaison des trois régimes précédents qui est observée.

Le transfert du gaz vers la surface solide, à travers la couche externe s'écrit :

$$N_g = k_d (C_e - C_s) \quad (4.8)$$

Où N_g : densité de flux du réactif ($\text{mol.m}^{-2}.\text{s}^{-1}$)

k_d : coefficient de transfert externe (m.s^{-1})

C_e : concentration du réactif dans la phase gaz (mol.m^{-3})

C_s : concentration du réactif à la surface du grain (mol.m^{-3})

La modélisation du réactif gazeux à travers la couche de cendres poreuse donne :

$$N_g = \frac{D_e R (C_s - C_c)}{R_0 (R_0 - R)} \quad (4.9)$$

Avec D_e : coefficient de diffusion à travers la couche de cendres ($\text{m}^2.\text{s}^{-1}$)

R_0 : rayon initial du grain (m)

R : rayon du cœur (m)

C_c : concentration du réactif à l'interface solide non réagi/gaz (mol.m^{-3})

Le point délicat de cette modélisation concerne la détermination du coefficient de diffusion au vu des paramètres intervenant dans son calcul (Annexe 3).

Enfin la dernière résistance concerne la réaction chimique au niveau du solide (grain de rayon R) :

$$N_g = k_i C_c \frac{R^2}{R_0^2} \quad (4.10)$$

Avec k_i : constante de vitesse de la réaction par unité d'aire interfaciale du front réactionnel (m.s^{-1})

En égalant les différentes expressions de la densité de flux, l'équation (4.11) est obtenue en posant l'hypothèse d'un ordre 1 pour la cinétique :

$$r_v = \frac{3}{R_0} C_e \frac{1}{\frac{1}{k_d} + \frac{R_0}{D_e} \left(\frac{1}{(1-X_A)^{1/3}} - 1 \right) + \frac{1}{k_i (1-X_A)^{2/3}}} \quad (4.11)$$

Où r_v : vitesse de transfert ($\text{mol.m}^{-3}.\text{s}^{-1}$)

X_A : taux d'avancement de la réaction

Remarque : Si l'hypothèse d'ordre 1 n'est pas remplie, alors la simplification avantageuse qu'elle procure à ce modèle n'est plus valable. Dans ce cas, il convient de changer de modèle.

Remarque : Grâce à des valeurs estimées lors d'études précédentes, le critère de Biot ($k_d L/D$, avec D la diffusion moléculaire) est très grand. Ce critère matérialise la facilité de diffusion du gaz dans le film gazeux par rapport à la diffusion dans les pores. Par conséquent le transfert externe sera négligé dans le cas présent.

Modélisation du lit fixe

Comme dans toute modélisation, avant d'aborder l'écriture des équations, il convient de poser quelques hypothèses sur le lit fixe :

- Le gaz traverse le lit fixe avec un écoulement piston mono-dimensionnel. Les conditions opératoires avec un écoulement à faible débit valident cette hypothèse.
- La diffusion (notamment la diffusion axiale due au gradient de concentration) est négligeable par rapport aux autres phénomènes.
- Les éléments de volume du lit sont isothermes (mais la température d'un élément peut évoluer au cours du temps), ce qui se justifie par le fait que la dispersion de la chaleur (axiale et radiale) est négligée tout comme les échanges thermiques solides/paroi (peu de contact entre une sphère et la paroi). De plus dans le dispositif expérimental, le réacteur est calorifugé, et donc il a peu de pertes thermiques à travers la paroi.

Ces hypothèses posées, les équations du modèle s'écrivent :

Bilans matières partiels :

$$-v_i r_v (1-\varepsilon) + \varepsilon \frac{\partial u_z C_i}{\partial z} + \varepsilon \frac{\partial C_i}{\partial t} = 0 \quad (4.12)$$

Avec : z : distance axiale du lit (m)

u_z : vitesse du gaz en fût vide (m.s^{-1})

C_i : concentration de l'espèce gazeuse i (mol.m^{-3})

ε : fraction de vide du lit (constante)

v_i : coefficient stœchiométrique de la réaction

Bilan matière global sur le solide :

$$\frac{\partial X_A}{\partial t} = \frac{vM}{\rho_s} r_v \quad (4.13)$$

Avec : X_a : avancement de la réaction

M : masse molaire du solide (kg.mol^{-1})

ρ_s : masse volumique du solide (kg.m^{-3})

Bilan thermique sur la phase fluide :

$$\rho C_p \frac{\partial u_z T_g}{\partial z} + \frac{(1-\varepsilon) 6h}{\varepsilon d_p} (T_g - T_s) + \rho C_p \frac{\partial T_g}{\partial t} = 0 \quad (4.14)$$

Avec : C_p : capacité calorifique massique du gaz ($\text{J.kg}^{-1}.\text{K}^{-1}$)

T_g : température du gaz (K)

T_s : température du solide (K)

ρ : masse volumique du gaz (kg.m^{-3})

h : coefficient d'échange thermique gaz/solide ($\text{W.m}^{-2}.\text{K}^{-1}$)

Bilan thermique sur le solide

$$\frac{6h}{d_p}(T_s - T_g) + r_v \Delta H_r + \rho_s C_p^s \frac{\partial T_s}{\partial t} = 0 \quad (4.15)$$

Avec : C_p^s : capacité calorifique massique du solide ($J.kg^{-1}.K^{-1}$)

ΔH_r : enthalpie de réaction ($J. mol^{-1}$)

Equation d'état, gaz parfait :

$$P = \sum_i C_i RT_g \quad (4.16)$$

Avec : P : pression (Pa)

R : constante des gaz parfaits ($J. mol^{-1}.K^{-1}$)

Pertes de charge (bilan mécanique) :

$$\frac{dP}{dz} = - \frac{f \rho u_z^2 \varepsilon^2}{d_p} \quad (4.17)$$

Avec : f : facteur de friction

Pour l'expression de la cinétique k_i suit la loi d'Arrhenius :

$$k_i = k_0 e^{-E_a / RT_s} \quad (4.18)$$

Avec : E_a : énergie d'activation ($J. mol^{-1}$)

k_0 : coefficient de cinétique ($m.s^{-1}$)

Ce modèle utilise de nombreuses données (masses volumiques, facteur de friction, capacités calorifiques....) qui sont évaluées par des corrélations ou des calculs. Le détail des corrélations utilisées est disponible en Annexe 3.

IV.3 Simulation et Résolution

Le problème d'identification paramétrique est basé sur la minimisation d'un critère obtenu par la méthode des moindres carrés :

$$\text{Min}_p \frac{1}{N} \left(\sum_{i=1}^N [x(t_i) - x_m(t_i, p)]^2 \right) \quad (4.19)$$

Avec : N : nombre de points d'échantillonnage

x : réponse expérimentale

x_m : réponse du modèle

p : paramètre à optimiser (identifier)

Avec ce critère, différentes contraintes sur l'ensemble des paramètres ou variables peuvent être ajoutées. Sur cet exemple, il n'a pas été nécessaire d'inclure ces contraintes. Compte tenu des informations disponibles à la suite de la campagne expérimentale, le critère prendra en compte 10 points de mesure (environ) et trois valeurs par point : flux d'hydrogène en sortie, température du solide en entrée et en sortie du réacteur. Initialement les trois paramètres à identifier sont k_0 , E_a et D_c (dans les cendres). L'ajout de ce dernier sert à pallier les incertitudes sur la tortuosité et le degré de vide.

Concernant la résolution, l'intégrateur est basé sur la méthode de Gear (chapitre 3) qui permet de traiter des systèmes d'équations différentielles et algébriques. Or le modèle précédent comporte des équations aux dérivées partielles. Par conséquent, les dérivées par rapport à la variable spatiale sont discrétisées. Au final, le réacteur est décomposé en une cascade de RAC en série. Cette discrétisation implique une multiplication du nombre d'équations. Ainsi le problème initial (un seul RAC) comporte 3 constituants (N_2 gaz inerte, H_2 , H_2O produite par la réaction) soit au total 10 équations. Au final le système comportera 10 équations qu'il faut multiplier par le nombre de points de discrétisation (nombre de RAC : entre 4 et 8 tout dépend de la longueur du lit).

Afin d'être au plus prêt des expériences, le déroulement de la simulation d'un essai est le suivant : au départ le réacteur contient uniquement que de l'azote, à $t=5s$ le gaz est mis en circulation avec une augmentation de la teneur en H_2 via un échelon.

IV.4 Résultats

Les valeurs d'initialisation des paramètres voués à l'identification, ont été choisies à partir d'observations expérimentales, et lors de tests du comportement du modèle ainsi que des valeurs issues de corrélations (Coefficient de diffusion). La figure 4.14 superpose les profils expérimentaux et les profils issus de la simulation pour les valeurs initiales des paramètres.

Les premiers résultats de l'identification font ressortir des difficultés pour s'approcher des résultats expérimentaux (à titre indicatif l'écart entre les pics sur T2 est de plus de 50 degrés). Ces dernières ont conduit à ajouter un paramètre de façon à diminuer le critère. Pour déterminer quel paramètre il faut soumettre à l'identification, deux voies sont explorées :

- Réaliser une analyse de sensibilité pour déterminer un paramètre ayant une influence importante sur le modèle.
- Regarder si les paramètres sont correctement évalués : utilisation d'une corrélation proche de ses limites de validité.

Au final, le coefficient d'échange thermique gaz-solide est apparu comme le plus adéquat. Tout d'abord le modèle est très sensible à ce paramètre, de plus pour les débits utilisés, l'évaluation de h par la corrélation retenue approche la limite de son domaine de validité.

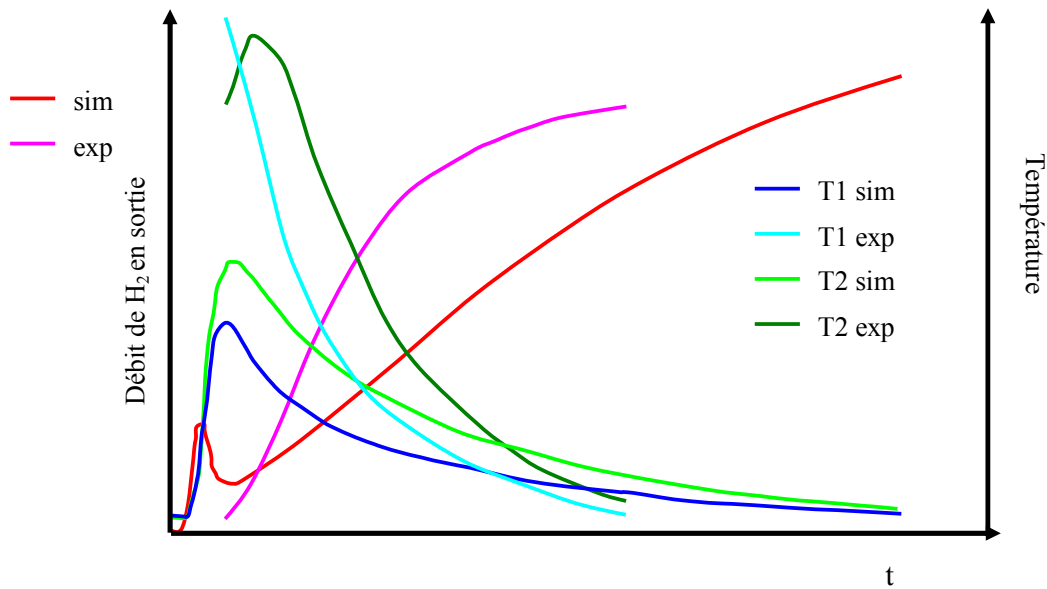


Figure 4.14 : Identification paramétrique initiale

Remarque : L'étude de la réaction a montré que des phénomènes peuvent modifier la diffusion dans le solide : apparition des cendres, formation d'eau, attrition... Pour formaliser cette évolution, le coefficient placé devant la corrélation de calcul de D_e est remplacé par une fonction linéaire de l'avancement X_A (Soit $AX_A + B$), ce qui entraîne l'identification d'un paramètre supplémentaire.

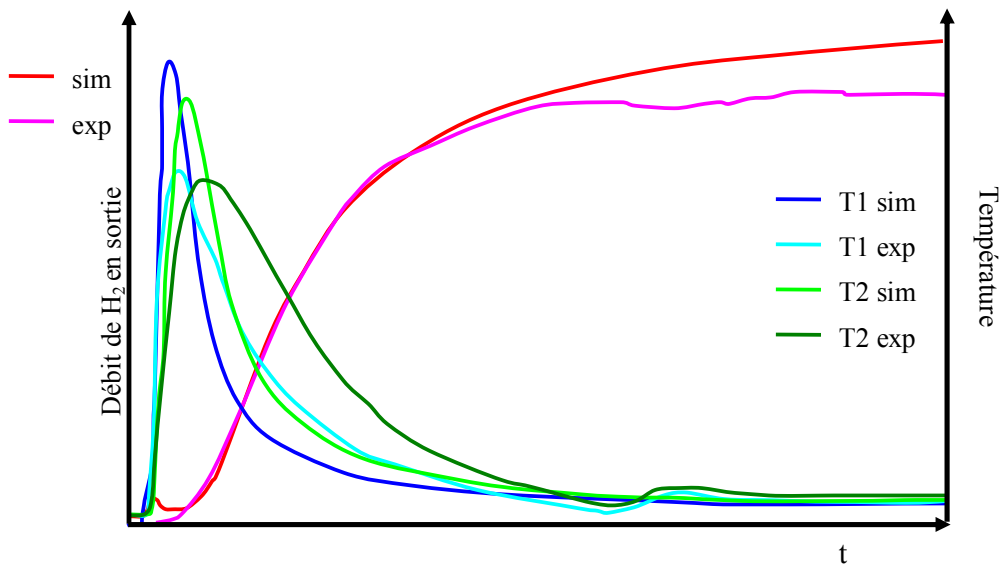


Figure 4.15 : Identification paramétrique après correction

Après l'identification, la simulation d'un essai de référence produit les résultats illustrés sur la figure 4.15. Sur la consommation d'hydrogène, les résultats numériques sont en adéquation avec les résultats expérimentaux jusqu'à la fin de la zone 2 (figure 4.13) (l'écart sur les pics de T2 n'est plus que de 20 degrés). Ensuite le phénomène présent dans la zone 3 ne semble pas être pris en compte : à savoir l'adsorption supplémentaire d'hydrogène observée dans cette zone. Il nécessite une modélisation particulière.

Les écarts sur les courbes de température sont plus conséquents à cause notamment de difficultés de modélisation des échanges. Ces difficultés sont dues : à la géométrie des grains, évolutions des grains qui peuvent perturber l'écoulement et les transferts. Ainsi le coefficient de transfert de chaleur h est susceptible de varier en cours d'essai.

V Conclusion

Ce chapitre a permis, au travers de trois applications, d'éclairer le lecteur sur la méthode implantée dans PrODHyS pour l'optimisation dynamique de procédé. Ces applications ont fait soulever trois points essentiels :

- La souplesse laissée à l'utilisateur grâce à ses divers choix notamment sur : la paramétrisation des fonctions de contrôle, la discrétisation de l'horizon de temps, le choix de la discrétisation du contrôle. Cette méthode possède également l'avantage de générer automatiquement les contraintes de continuité et surtout de pouvoir facilement inclure les contraintes sur les variables de contrôle : comme le démontrent les deux premiers exemples.
- Le second exemple, quant à lui met clairement en évidence une des limites de l'approche. En effet la prise en compte de contraintes sur les fonctions d'état nous oblige : soit à inclure des variables supplémentaires et de leurs contraintes associées au temps final, soit, si le cas est favorable, à augmenter le nombre de points de discrétisation de l'horizon de temps (se traduisant par une augmentation du temps calcul).
- Enfin le dernier exemple, démontre l'étendue de l'ajout de la partie optimisation à cette plateforme. Grâce à ce couplage simulation – optimisation, il est désormais facile de pouvoir faire de l'identification paramétrique en régime dynamique avec PrODHyS. Les résultats dans le cadre d'un lit fixe en témoignent.

Conclusion

Ces dernières années, les systèmes dynamiques hybrides ont reçu une attention particulière tout domaine confondu. En effet, les procédés batchs, les systèmes de transport, les protocoles de communication, n'ont fait qu'alimenter la nécessité de faire évoluer ce domaine d'études. Les recherches menées sont articulées autour de trois axes principaux : la modélisation, l'analyse et la commande. Pour notre part, l'objectif visé était d'aborder le problème d'optimisation des systèmes dynamiques hybrides. Le terme d'optimisation des systèmes dynamiques hybrides peut faire référence à deux problèmes différents qui correspondent chacun à un niveau d'optimisation différent :

- Le premier concerne le problème d'optimisation de la (ou des) partie continue, plus communément dénommé : problème de contrôle optimal.
- Pour le second type de problème, l'optimisation se situe à un niveau plus élevé, car elle concerne l'optimisation de l'ordre des séquences continues. Le but étant alors de déterminer le séquençement optimal des différents modes.

Ce travail de recherche est dédié au premier type de problème, à savoir le problème de contrôle optimal.

Cette étude a choisi comme domaine d'application le génie des procédés. Dans ce domaine, la modélisation dite « hybride » est d'une grande importance car elle découle du besoin de modéliser des systèmes réels de plus en plus complexes mais dont la simplification des dynamiques conduit à des discontinuités dans la représentation. Pour les procédés batchs, les phases transitoires des opérations unitaires (démarrages, arrêts, nettoyage), les changements d'états physiques de la matière en sont de parfaits exemples.

Dans un premier temps, ce travail s'est centré sur les modèles et les outils de simulation couramment rencontrés dans le domaine des systèmes dynamiques hybrides. Dans ce cadre, il était important, de répertorier les propriétés permettant de les caractériser. Les modèles se définissent en termes de modularité, de causalité, de représentation explicite rigoureuse des composantes continues et discrètes, de capacités à gérer la structure variable du système algébro-différentiel global. Les outils de simulation sont caractérisés par le caractère modulaire ou global de la simulation, leurs capacités à traiter les équations différentielles ordinaires / partielles / algébriques, la possession d'un algorithme de réduction d'index, la gestion des événements, la réinitialisation explicite et implicite des variables.

Ensuite, avant de mettre en œuvre la méthode retenue, ce manuscrit dresse un état de l'art des méthodes couramment rencontrées dans le domaine de l'optimisation dynamique. Ces dernières sont classées selon deux grandes catégories : les méthodes indirectes et les méthodes directes. Les premières découlent de l'application du principe du maximum de Pontryagin. Quant aux secondes, elles transforment le problème de contrôle optimal en un problème d'optimisation non linéaire en dimension finie. Dans cette seconde catégorie de méthode, une autre classification est réalisable en fonction du niveau de discrétisation retenue : discrétisation partielle du problème (approche séquentielle) ou discrétisation totale (approche simultanée).

La méthode retenue et implémentée peut être considérée comme une méthode intermédiaire entre les deux approches précédentes. Dans le principe, cette méthode partitionne l'horizon de temps en sous intervalles où le système d'équations différentielles et algébriques est intégré.

Les points forts de cette méthode sont divers :

- Tire profit des fonctionnalités de l'intégrateur DISCo
- Prend en compte des contraintes sur les variables d'état
- N'augmente pas trop la taille des problèmes
- Permet la parallélisation des calculs

La méthode ayant été clairement présentée, elle a ensuite été insérée au sein de la plateforme de simulation dynamique PrODHyS. Dans la dernière partie de ce manuscrit, cette implantation a permis d'utiliser la méthode évoquée, au travers de trois exemples d'application. Le premier, concernant un réacteur fed batch, a montré la démarche globale que l'utilisateur doit adopter pour réaliser son optimisation. Cet exemple simple mais aussi en évidence, la prise en compte de contraintes uniquement sur les fonctions de contrôle. Le deuxième, relatif à un réacteur batch avec coulée, a permis d'illustrer les limites de la méthode avec la difficulté engendrée par la prise en compte de contraintes sur les variables d'états. Enfin, le troisième a mis l'accent sur le caractère extensible de la méthode, avec une identification paramétrique en mode discontinu.

Ces travaux de recherche ouvrent de nombreuses perspectives pour l'avenir dont notamment :

- Dans le chapitre 3, lors de la présentation de la méthode de nombreuses voies d'amélioration de celle-ci ont été proposées : comme par exemple le choix de la méthode pour résoudre le système EDA global. Pour l'instant, seule une méthode est disponible, cependant on a souligné que la méthode à correction décalée était souvent plus performante (méthode 3). Au travers de cet exemple, il est clairement visible que la version actuelle de l'outil est perfectible avec l'ajout d'options supplémentaires pour l'utilisateur voire d'amélioration des performances de ce dernier.
- Le deuxième exemple du chapitre 4 a clairement mis en évidence une des limites de cette approche. En effet, la prise en compte de contraintes sur les variables d'états s'avère difficile. Elle est possible grâce à l'introduction de variables supplémentaires avec leurs contraintes associées au temps final, ou grâce à une discrétisation plus fine de l'horizon de temps au détriment du temps calcul.
- Ce travail se concentre uniquement sur l'optimisation de la partie continue de la simulation dynamique. Or, les procédés peuvent également comporter une composante discrète, qui couplée à la composante continue forme des Systèmes Dynamiques Hybrides. Dans les procédés discontinus, des recettes de fabrication sont effectuées étapes par étapes. Dans ces ateliers, plusieurs recettes utilisent les mêmes appareils et peuvent être réalisées en parallèle. De plus chaque recette comporte plusieurs étapes à effectuer en parallèle ou en série. Au final, une voie d'amélioration possible est de mettre en place une méthode pour optimiser l'enchaînement de ces étapes afin d'avoir une optimisation complète (partie continue et partie discrète) des systèmes dynamiques hybrides.

ANNEXES

Annexe 1

Exemple de Bond Graph

L'exemple présenté au chapitre 2 issu de Couenne et al (2006a), est étendu en considérant le fluide comme visqueux. Sa température est supposée uniforme et égale à T et la température extérieure est notée T_0 .

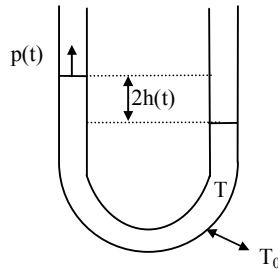


Figure A1.1 : Exemple du chapitre 2, tube en U.

Les frottements provoquent une élévation de température dans le liquide et qui induit un flux de chaleur entre le liquide et le milieu extérieur. La description de cet échange sur le bond graph, nécessite l'ajout d'un nouvel élément C associé à l'énergie interne U , du liquide. Il faut définir des couples de variables conjuguées. En thermodynamique, c'est l'équation de Gibbs qui exprime la différentielle de U (considérée comme une fonction de l'entropie S , du volume V , et du nombre de mole d'un constituant i).

$$dU = TdS - PdV + \sum_i \mu_i dn_i$$

Avec μ_i le potentiel chimique de i .

L'élément C associé à U est un élément multi ports puisqu'il peut échanger de l'énergie de trois manières : énergie thermique, énergie associée au changement de volume, énergie due au transfert. Dans l'exemple présenté, seule l'énergie thermique est considérée, par hypothèse. Le bond graph de l'exemple étendu est illustré sur la figure A1.2.

Le bilan d'espace associé à l'énergie potentielle reste inchangé, par contre celui sur la quantité de mouvement est modifié à cause de la présence des forces de viscosité F_{vis} . Ces deux bilans, représentent par des éléments 0 (effort commun et somme des flux nulle) :

Bilan de quantité de mouvement (Energie Cinétique) :
$$\frac{d(\rho l dp)}{dt} = -\rho g d 2h - F_{vis}$$

Bilan d'espace (Energie potentielle) :
$$\frac{dh}{dt} = v$$

Il reste à exprimer le bilan d'entropie. L'application du second principe exprime que l'entropie n'est pas conservative. Des termes sources positifs associés aux phénomènes irréversibles doivent être pris en compte dans le bilan :

$$\text{Bilan d'entropie (énergie interne)} : \frac{dS}{dt} = f_s^0 + \sigma_{vis} + \sigma_{therm} = \frac{\Phi_0}{T_0} + \frac{F_{vis} v}{T} + \Phi_0 \left(\frac{1}{T} - \frac{1}{T_0} \right)$$

Le terme $f_s^0 = \frac{\Phi_0}{T_0}$ représente le flux d'entropie échangé avec le milieu extérieur à la température T_0 fixée. T_0 fixe, est représentée sur le graph par l'élément Se qui signifie source d'effort. En continuant le parcours de l'énergie thermique (à partir de la source), un élément l est rencontré (flux commun et somme des efforts nulle) : il permet de définir une variable d'effort $(T_0 - T)$ jouant le rôle de force motrice pour le transfert thermique. Deux phénomènes dissipatifs ou irréversible, traduits par deux éléments R, se produisent. Ces éléments permettent de définir une relation entre une variable de flux et une variable d'effort :

- pour le transfert thermique, le flux de chaleur (flux d'entropie) est proportionnel à $(T_0 - T)$, via un coefficient d'échange.
- La force de frottement est reliée à la vitesse via la viscosité.

De plus ces éléments R représentent le flux d'entropie créée par le phénomène irréversible représenté, associé à l'effort correspondant, de telle façon que le produit des deux donne la puissance perdue à cause de ces irréversibilités :

$$T \sigma_{vis} = F_{vis} v$$

$$T \sigma_{therm} = f_s^0 (T_0 - T) = \frac{\Phi_0}{T_0} (T_0 - T)$$

Finalement, il faut vérifier que le bilan instantané d'énergie du système est bien satisfait :

$$\frac{d(E_{tot} + U)}{dt} = v \frac{d(\rho l dp)}{dt} + \rho g d 2h \frac{dh}{dt} + T \frac{dS}{dt} = \Phi_0$$

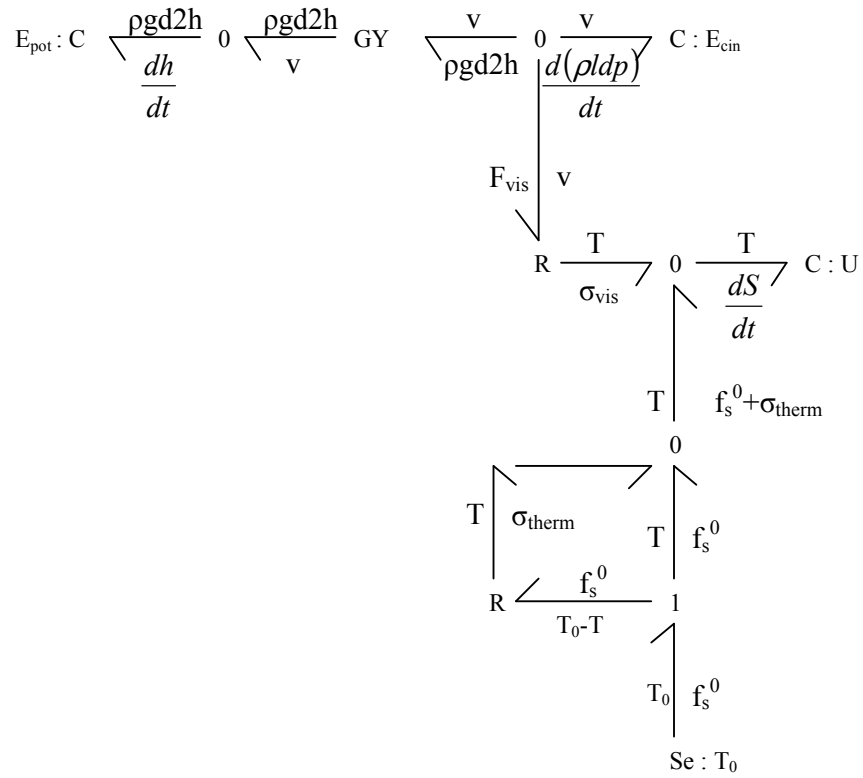


Figure A1.2 : Représentation du bond graph avec échange thermique

Annexe 2

Algorithme Général de la méthode PDI :

- Etape 1 : Diviser l'intervalle de temps en P sous intervalles de même longueur L.
- Etape 2 : Choisir le nombre de point de maillage, N ainsi que le nombre de valeur possible pour le contrôle M. Choisir également la région initiale de contrôle r, tout en tenant compte d'éventuelles contraintes sur cette région.
- Etape 3 : Choisir les M valeurs pour la fonction de contrôle (ou une puis génération automatique des M-1 autres).
- Etape 4 : Intégrer le système EDA en utilisant les M valeurs pour les fonctions de contrôle ainsi générer et stocker à chaque point de temps les N points pour le vecteur d'état (point de maillage).
- Etape 5 : Partant du dernier intervalle, correspondant aux temps t_{f-L} à t_f , pour chaque point de maillage intégrer le système EDA de t_{f-L} à t_f pour chaque M^m valeurs possibles pour le contrôle (ou des R valeurs dans le cas d'un tirage aléatoire). Puis pour chaque point de maillage retenir la trajectoire qui minimise la fonction objectif et stocker la valeur retenue pour le contrôle (qui servira dans l'étape suivante).
- Etape 6 : Passer à l'intervalle P-1 correspondant du t_{f-2L} à t_{f-L} . De même pour chaque point de maillage intégrer le système EDA de t_{f-2L} à t_{f-L} avec les M^m valeurs possibles pour le contrôle (ou des R valeurs dans le cas d'un tirage aléatoire). Pour continuer l'intégration de t_{f-L} à t_f choisir la trajectoire obtenue à l'étape 5 correspondant au point de maillage le plus proche. Puis pour chaque point de maillage retenir la trajectoire qui minimise la fonction objectif et stocker la valeur retenue pour le contrôle (qui servira dans l'étape suivante).
- Etape 7 : Répéter la procédure pour tous les intervalles jusqu'au premier. Au final stocker la politique de contrôle qui minimise la fonction objectif.
- Etape 8 : Réduire la taille de la région des contrôles admissibles r en lui appliquant un facteur de contraction : $r^{j+1} = \gamma' r^j$ avec $0,5 \leq \gamma' < 1$. Utiliser la politique de

contrôle du pas précédent comme valeur de référence pour générer les M-1 autres politiques possibles.

Etape 9 : Augmenter le nombre d'itération de 1 et reprendre l'algorithme à l'étape 3. Continuer la procédure pour un certains nombres d'itérations et examiner le résultat en enregistrant la valeur du critère et la meilleure politique de contrôle pour chaque itération.

Annexe 3

Données nécessaires pour le modèle.

Certaines données sont calculées durant la simulation à partir de la bibliothèque BibPhy du logiciel ProPhy qui a été couplée au programme de simulation. Toutefois, l'ensemble des données n'est pas calculé de cette manière, le programme fait également appel à des données calculées par corrélations.

Masse Volumique solide

La masse volumique du solide est déduite de sa masse et du volume du lit fixe :

$$\rho_s = \frac{m_s}{(1-\varepsilon)V_1} \quad \text{où} \quad V_1 = L\pi \frac{d_t^2}{4} \quad (\text{A3.1})$$

Avec m_s : masse de solide dans le lit (kg)

L : longueur du lit fixe (m)

d_t : diamètre du lit (m)

Capacité calorifique solide

La capacité calorifique du solide est évaluée par une corrélation en fonction de la température :

$$C_p^s = A + BT + CT^2 \quad (\text{A3.2})$$

Où A,B et C sont des constantes, non données ici pour cause de confidentialité sur l'oxyde métallique retenu.

Coefficient de diffusion dans les cendres

Le calcul du coefficient de diffusion est délicat du fait du manque d'informations sur la structure du solide. La méthode d'évaluation du coefficient dépend de phénomènes prépondérants : collisions intermoléculaires dans le gaz ou entre les molécules et la paroi des pores (diffusion de Knudsen). Il est déterminé en calculant le libre parcours moyen des molécules d'hydrogène et en les comparant à la taille des pores. Cette dernière sera supposée de l'ordre de 20 Angström (diamètre généralement admis pour ce type de solide, noté d_p). Le libre parcours moyen est ainsi calculé :

$$l = \frac{k_B T}{\sqrt{2P\pi R_{H_2}^2}} \quad (\text{A3.3})$$

Avec : $R_{H_2}^2$: rayon de l'atome d'hydrogène, 1.17 Angström

k_b : constante de Boltzmann $1.28 \cdot 10^{-23} \text{ J.K}^{-1}$

P : la pression (Pa)

T : température (K)

A 70°C, sous 1 bar, $l=1.9 \cdot 10^{-7}$ d'où $d_p/l=200/1900 \approx 0.1$ ce qui nous place dans le domaine de prépondérance de la diffusion de Knudsen. Le coefficient s'écrit alors :

$$D_k = 1.534 d_p \sqrt{\frac{T_g}{M_{H_2}}} \quad (\text{A3.4})$$

Avec : d_p : diamètre des pores (m)

T_g : température du gaz (K)

M_{H_2} : masse molaire ($\text{kg} \cdot \text{mol}^{-1}$)

Le coefficient de diffusion effectif est obtenu en appliquant le modèle des pores tortueux afin de tenir compte de la complexité de la géométrie du système des pores. Ainsi, l'introduction du degré de vide interne β et de la tortuosité τ donnent :

$$D_e = \frac{\beta D_k}{\tau} \quad (\text{A3.5})$$

Avec : β compris entre 0.4 et 0.5

τ compris entre 2 et 8

En pratique on prendra $\beta=0.45$ et $\tau=4$ mais pour tenir compte de ces incertitudes la valeur sera pondérée par un paramètre soumis à l'identification paramétrique (chapitre 4).

Coefficient d'échange thermique

Le coefficient d'échange thermique solide/gaz est déterminé à partir de corrélations. Pour des valeurs du nombre de Reynolds supérieurs à 100 et des diamètres de particules supérieurs à 1 mm :

$$Nu = \frac{h d_p}{\lambda_g} = 2.0 + 1.8 \text{Pr}^{0.33} \text{Re}^{0.5} \quad (\text{A3.6})$$

Avec : h : coefficient d'échange thermique ($\text{W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$)

λ_g : conductivité thermique du gaz ($\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$)

Re : nombre de Reynolds, $\text{Re} = \frac{\rho_g u_z d_p}{\mu_g}$

ρ_g : masse volumique du gaz

u_z : vitesse du gaz

μ_g : viscosité dynamique du gaz

Pr : nombre de Prandtl, $\text{Pr} = \frac{C_p^g \mu_g}{\lambda_g}$

C_p^g : capacité calorifique du gaz

Pour des diamètres de particules inférieurs à 1 mm, la détermination d'une corrélation devient plus délicate et les corrélations présentées sortent de leur domaine de validité. Ce qui explique le choix de ce coefficient comme paramètre soumis à l'identification. Pour des valeurs de Re inférieures à 100 (ce qui est possible lors des essais expérimentaux), la relation suivante permet d'avoir un ordre de grandeur :

$$Nu = 0.03 Re^{1.3} \quad (A3.7)$$

Enthalpie de réaction

L'enthalpie de réaction ΔH_r est calculée à 25°C à partir des enthalpies de formation des produits intervenants, en les pondérant par les coefficients stœchiométriques :

$$\Delta H_r(25^\circ C) = \sum_i \nu_i \Delta H_{f,i}^0 \quad (A3.8)$$

Pour obtenir cette enthalpie à une température T :

$$\Delta H_r(T) = \Delta H_r(25^\circ C) + \int_{25^\circ C}^T C_p(T) dT \quad (A3.9)$$

Taux de vide du lit

Le taux de vide du lit est évalué par la corrélation de Haughey et Beveridge en supposant les grains sphériques.

$$\varepsilon = 0.38 + 0.073 \left(1 + \frac{(d_t / d_p - 2)^2}{(d_t / d_p)^2} \right) \quad (A3.10)$$

Facteur de friction

Il est déterminé par la corrélation suivante :

$$f = \frac{1 - \varepsilon}{\varepsilon^3} \left(A + B \frac{1 - \varepsilon}{Re} \right) \quad (A3.11)$$

Avec : A=1.75 et B=150 si $\frac{Re}{1 - \varepsilon} \leq 500$ (Ergun)

A=1.24 et B=368 si $1000 \leq \frac{Re}{1 - \varepsilon} \leq 5000$ (Handley et Hegg)

Au-delà de $\frac{Re}{1 - \varepsilon} = 5000$ la relation de Hicks est utilisée :

$$f = 6.8 \frac{(1 - \varepsilon)^{1.2}}{\varepsilon^3 Re^{0.2}} \quad (A3.12)$$

Références Bibliographiques

Bibliographie

A

Afnor, *Diagramme fonctionnel GRAFCET pour la description des systèmes logiques de commande*, Norme Française NF C03-190, Juin, 1982.

Albuquerque J., Biegler L., *Decomposition algorithms for on line estimation with non linear DAE models*, Computers and Chemical Engineering, 21 (3), pp 283-299, 1997.

Albuquerque J., Gopal V., Staus G., Biegler L., Ydstie B., *Interior point SQP strategies for large scale structured process optimization problems*, Computers and Chemical Engineering, 23, pp 543-554, 1999.

Allgower E. L., Georg K., *Numerical Continuation Methods : An introduction, Series in Computational Mathematics*, 13, Springer, Berlin, 1990.

Allgower E. L., Georg K., *Piecewise linear methods for nonlinear equations and optimization*, Journal of Computational and Applied Mathematics, 124 (1-2), pp 245-261, 2000.

Alur R., Courcoubetis C., Halbwachs N., Henzinger T.A., Ho P.H., Nicollin X., Olivero A., Sifakis J., Yovine S., *The algorithmic analysis of hybrid systems*, Theoretical Computer Science, 138, pp 3-34, 1995.

Alur R., Dill D.L., *A theory of timed automata*, Theoretical Computer Science, 126 (2), pp 183-225, 1994.

B

Bathia T., Biegler T., *Dynamic Optimization in the Design and Scheduling of Multiproduct Batch Plants*, Industrial & Engineering Chemistry Research, 35, pp 2234-2246, 1996.

Belaud J.P., Pons M., *Open software architecture for process simulation : the current status of Cape-Open standard*, European Symposium of Computer Aided Process Engineering (ESCAPE 12), The Hague, Pays Bas, 2002.

Bell M.L., Limebeer D.J.N., Sargent R.W.H., *Robust receding horizon optimal control*, Computers and Chemical Engineering, Suppl., 20, ppS781-S786, 1996.

Betts J., *Practical methods for optimal control using nonlinear programming, Advances in design and control*, SIAM, Philadelphie, 2001.

Biegler L.T., *Improved infeasible path optimization for sequential modular simulators—I: The interface*, Computers & Chemical Engineering, 9 (3), pp 245-256, 1985.

Biegler L.T., *An overview of simultaneous strategies for dynamic optimization* (à paraître).

Biegler L.T., Cervantes A., Wächter A., *Advances in simultaneous strategies for dynamic process optimization*, Chemical Engineering Science, 57, pp 575-593, 2002.

Biegler L.T., Grossmann I.E., *Retrospective on optimization*, Computers and Chemical Engineering, 28, pp 1169-1192, 2004.

Bilous O., Amundson N.R., *Optimum temperature gradients in tubular reactors – I. General theory and methods*, Chemical Engineering Science, 5, pp 81-92, 1956a.

Bilous O., Amundson N.R., *Optimum temperature gradients in tubular reactors – II. Numerical study*, Chemical Engineering Science, 5, pp 115-126, 1956b.

Bloss K.F., Biegler L.T., Schiesser W.E., *Dynamic process optimization through adjoint formulations and constraint aggregation*, Industrial & Engineering Chemistry Research, 38, pp 421-432, 1999.

Bock H., Platt K., *A multiple shooting algorithm for direct solution of optimal control problem*. In 9th IFAC world congress, Budapest, pp 242-247, 1984.

Bojkov B., Luus R., *Use of random admissible values for control in iterative dynamic programming*, Industrial & Engineering Chemistry Research, 31, pp 1308-1314, 1992.

Bojkov B., Luus R., *Time optimal control by iterative dynamic programming*, Industrial & Engineering Chemistry Research, 33, pp 1486-1492, 1994.

Bonnard B., Faubourg L., Launay G., Trélat E., *Optimal control with state constraints and the space shuttle re-entry problem*, Journal of Dynamical and Control Systems, 9 (2), pp 155-199, 2003.

Buisson J., Cormerais H., *Descriptor systems for the knowledge modeling and simulation of hybrid physical systems*, Journal Européen des Systèmes Automatisés, 32 (9-10), pp 1047-1072, 1998.

C

Cao Y., Li S., Petzold L., *Adjoint Sensitivity for differential algebraic equations algorithms and software*, Journal of Computer and Applied Mathematics, 149, pp 171-192, 2002.

Cao Y., Li S., Petzold L., Serban R., *Adjoint sensitivity for differential-algebraic equations: the adjoint DAE system and its numerical solution*, Society for Industrial and Applied Mathematics Journal on Scientific Computing, 24 (3), pp 1076-1089, 2003.

CEI-IEC, *Preparation of function charts for control systems*, Norme Internationale CEI 848, 1988.

Cervantes A., Biegler L., *Large-Scale DAE Optimization Using a Simultaneous NLP Formulation*, AIChE Journal, 44 (5), pp 1038-1050, 1998.

Cervantes A., Biegler L., *A stable elemental decomposition for dynamic process optimization*, Journal of Computational and Applied Mathematics, 120, pp 41-57, 2000.

Cervantes A., Tonelli S., Brandolin A., Bandoni A., Biegler L., *Large Scale dynamic optimization of a low density polyethylene plant*, Computer and Chemical Engineering, 24, pp 983-989, 2000 b.

Cervantes A., Tonelli S., Brandolin A., Bandoni A., Biegler L., *Large Scale dynamic optimization for grade transitions in a low density polyethylene plant*, Computer and Chemical Engineering, 26, pp 227-237, 2002.

Cervantes A., Wächter A., Tütüncü R., Biegler L., *A reduced space interior point strategy for optimization of differential algebraic systems*, Computers and Chemical Engineering, 24, pp 39-51, 2000 a.

Ceynowa J., Adamczak P., *Analysis of the bond graph network model of membrane reactor for olive oil hydrolysis*, Separation and Purification Technology, 22-23, pp 443-449, 2001.

Champagnat R., *Supervision des systèmes discontinus : définition d'un modèle hybride et pilotage en temps réel*, Thèse de Doctorat, LAAS Toulouse, France, 1998.

Champagnat R., Andreu D., Pingaud H., Pascal J.C., Esteban P., *Modélisation des systèmes de production discontinue : proposition d'un modèle pour la supervision*, Rapport LAAS N° 97524, Journal Européen des Systèmes Automatisés, 32 (5-6), pp 725-756, 1998a.

Champagnat R., Pingaud H., Alla H., Valentin-Roubinet C., Flaus J.M., Valette R., *A gas storage example as a benchmark for hybrid modelling*, European Journal of Automation, 32 (9-10), 1998b.

Champagnat R., Pingaud H., Valette R., *A model for hybrid systems : differential predicate transitions Petri nets*, Rapport LAAS N° 99024, Toulouse, 1999.

Couenne F., Jallut C., Maschke B., Tayakout M., Breedveld P., *Les graphes de liaison pour la modélisation dynamique en génie des procédés*, SIMO 2006, Toulouse France, 2006a.

Couenne F., Jallut C., Maschke B., Breedveld P., Tayakout M., *Bond Graph modelling for chemical reactors*, Mathematical and Computer Modeling of Dynamical Systems, 12 (2-3), pp 159-174, 2006b.

Coward I., *The time-optimal problem in binary batch distillation*, Chemical Engineering Science, 22, pp 503-516, 1967a.

Coward I., *The time-optimal problem in binary batch distillation – A further note*, Chemical Engineering Science, 22, pp 1181-1184, 1967b.

Cuthrell J., Biegler L., *On the optimization of differential-algebraic process systems*, AIChE Journal, 33, pp 1257-1270, 1987 AIChE Journal, 45 (4), pp 781-801, 1999.

D

Dadebo S.A., Mc Auley K.B., *Dynamic Optimization of constrained chemical engineering problems using dynamic programming*, Computers and Chemical Engineering, 19 (5), pp 513-525, 1995.

Denbigh K., *Velocity and yield in continuous reactions systems*, Transactions of the Faraday Society, 40, pp 352-373, 1944.

E

Elgue S., *Optimisation des synthèses pharmaceutiques globales intégrant les aspects environnementaux et de sécurité*, Thèse de Doctorat INP Toulouse, France, 2002.

F

Farhats S., Czernicki M., Pibouleau L., Domenech S., *Optimization of multiple-fraction batch distillation by nonlinear programming*, AIChE Journal, 36 (9), pp. 1349-1360, 1990.

Feehery W.F., Tolsma J.E., Barton P.I., *Efficient sensitivity analysis of large-scale differential-algebraic systems*, Applied Numerical Mathematics, 25, pp 41-54, 1997.

Feehery W.F., Barton P.I., *Dynamic optimization with state variable path constraints*, Computers and Chemical Engineering, 22 (9), pp 1241-1256, 1998.

Feehery W.F., Barton P.I., *Dynamic optimization with equality path constraints*, Industrial & Engineering Chemistry Research, 38, pp 2350-2363, 1999.

Fikar M., Latifi A., Fournier F., Creff Y., *Control vector parametrisation versus Iterative dynamic programming in dynamic optimisation of a distillation column*, Computers and Chemical Engineering, 22 (Suppl), pp S625-S628, 1998.

Flaus J.M., *Modeling Analysis of hybrid dynamical systems : an introduction*, Journal Européen des Systèmes Automatisés, 32 (7-8), pp 797-830, 1998.

Furlonge H.I., Pantelides C.C., Sorensen E., *Optimal operation of multivessel batch distillation columns*, AIChE Journal, 45 (4), pp 781-801, 1999.

G

Gani R., Braunschweig B., *Software architectures and tools for computer aided process engineering*, Elsevier, ISBN 0-444-50827-9, 2002.

Gear C.W., *The simultaneous Numerical solution of differential-algebraic equations*, IEEE Transaction on Corcuit theory, CT 18 (1), Ed. Academic Press, 1971.

Goh C., Theo K., *Control parameterization : a unified approach to optimal control problem with general constraints*, Automatica, 28, pp 3-18, 1988.

H

Harel D., *Statecharts: a visual formalism for complex system*, Science of computer programming, 8, pp 231-274, 1987.

Hartl R.F., Sethi S.P., Vickson R.G., *A survey of the maximum principles for optimal control problems with state constraints*, SIAM Review, 37 (2), pp 181-218, 1995.

Hopcroft J.E., Ullman J.D., *Introduction to Automata theory, Languages and Computation*, Addison-Wesley, 1979.

I

Ishikawa T., Natori Y., Liberis L., Pantelides C.C., *Modeling and optimization of an industrial batch process for the production of dioctyl phthalate*, Computers and Chemical Engineering, Suppl., 21, pp S1239-S1244, 1997.

J

Jackson R., Obando R., *On the control of competing chemical reactions*, Chemical Engineering Science, 26, pp 853-865, 1971.

Jackson R., Senior G., *The optimization of chemical reactors by the delayed addition of reactant*, Chemical Engineering Science, 23, pp 971-980, 1968.

Jensen K., Rozenberg G., *High level Petri nets : theory and application*, Springer Verlag, ISBN 3-540-54125-x, 1991.

Jockenhövel T., Biegler L., Wätcher A., *Dynamic optimization of the Tennessee Eastman process using the OptControlCentre*, Computer and Chemical Engineering, 27, pp 1513-1531, 2003.

Jourda L., *Composants logiciels orientés objets pour la modélisation et la simulation des procédés chimiques*, Thèse de Doctorat INP Toulouse, France, 1996.

Jourda L., Joulia X., Koehret B., *Introducing ATOM, the Applied Thermodynamic Object Model*, Computers and Chemical Engineering, Suppl. A, 20, ppS157-S164, 1996.

K

Kao C., *Performance of several non linear programming software packages on microcomputers*, Computers and Operations Research, 25 (10), pp 807-816, 1998.

Kameswaran S., Biegler L., *Simultaneous Dynamic Optimization Strategies : Recent Advances and Challenges*, Computers and Chemical Engineering, 30, pp 1560-1575, 2006.

Karnopp D., Rosenberg R., *Systems dynamics. A unified approach*, Wiley Intersciences, New York, 1996.

Kerkhof L., Vissers H., *On the profit of optimal control in batch distillation*, Chemical Engineering Science, 33, pp 961-970, 1978.

Kesten Y., Pnueli A., *Timed and hybrid statecharts and their textual representation*, Lecture Notes in Computer Science, 571, 1992.

L

Lakshmanan A., Rooney W., Biegler L., *A case study reactor network synthesis : the vinyl chloride process*, Computers and Chemical Engineering, 23, pp 479-495, 1999.

Le Lann J.M., *Habilitation à diriger des Recherches*, INP Toulouse, France, 1999.

Leineweber D., Bauer I., Bock G., Schlöder J., *An efficient multiple shooting based reduced SQP strategy for large scale dynamic process optimization. Part I : theoretical aspects*, Computers and Chemical Engineering, 27, pp 157-166, 2003 a.

Leineweber D., Bauer I., Bock G., Schlöder J., *An efficient multiple shooting based reduced SQP strategy for large scale dynamic process optimization. Part II : software aspects and applications*, Computers and Chemical Engineering, 27, pp 167-174, 2003 b.

Li S., Petzold L., Zhu W., *Sensitivity analysis of differential-algebraic equations : a comparison of methods on a special problem*, Applied Numerical Mathematics, 32, pp 161-174, 2000.

Li S., Petzold L. *Description of DASPKADJOINT: an adjoint sensitivity solver for differential algebraic equations*, Technical Report, 2003.

Luus R., *Optimal control by dynamic programming using accessible grid points and region contraction*, Hungarian Journal of Industrial Chemistry, 17, pp 523-543, 1989.

Luss R., Rosen O., *Application of dynamic programming to final state constrained optimal control problems*, Industrial & Engineering Chemistry Research, 30, pp 1525-1530, 1991.

Luus R., *Piecewise linear continuous optimal control by iterative dynamic programming*, Industrial & Engineering Chemistry Research, 32, pp 859-865, 1993.

Logsdon J. S., Biegler L., *Accurate solution of differential-algebraic optimization problems*, Industrial & Engineering Chemistry Research, 28 (11), pp 1628-1639, 1989.

M

Maly T., Petzold L., *Numerical methods and software for sensitivity analysis of differential-algebraic systems*, Applied Numerical Mathematics, 20, pp 57-79, 1996.

Martinon P., *Résolution numérique de problèmes de contrôle optimal par une méthode homotopique simpliciale*, Thèse de Doctorat INP Toulouse, France, 2005.

Mattsson S., Olsson H., Elmqvist H., *Dynamic selection of states in Dymola*, Modelica workshop 2000 proceedings, pp 61-67, 2000.

Mayur D., Jackson R., *Time-optimal problems in batch distillation for multicomponent mixtures and for columns with holdup*, Chemical Engineering Journal, 2, pp 150-163, 1971.

Miele A., *Recent advances in the optimization and guidance of aeroassociated orbital transfers*, Acta Astronautica, 38 (10), pp 747-768, 1996.

Mekarapiruk W., Luus R., *Optimal control of inequality state constrained systems*, Industrial & Engineering Chemistry Research, 36, pp 1686-1694, 1997.

Montealegre M., *Intégration du design et de la commande optimale : application à la distillation réactive*, Thèse de Doctorat Université de Pau et des Pays de l'Adour, France, 2005.

Moyse A., *Modélisation orientée objet des opérations unitaires*, Thèse de Doctorat INP Toulouse, France, 2000.

Mujtaba I.M., Machietto S., *Efficient optimization of batch distillation with chemical reaction using polynomial curve fitting techniques*, Industrial & Engineering Chemistry Research, 36, pp 2287-2295, 1997.

O

Ould Bouamama B., Dauphin-Tanguy G., *Modélisation Bond Graph Elément de base pour l'énergétique*, Technique de l'ingénieur, BE 8 280, 2005.

P

Pagès A., Pingaud H., *An hybrid process model based on Petri nets applied to short term scheduling of batch-semi continuous plants*, Proceeding of the Workshop on analysis and design of event-driven operations in process systems (ADEDOPS), Imperial College, London, UK, 1995.

Pantelides C., *The consistent initialization of differential-algebraic systems*, SIAM Journal on Scientific and statistical computing, 9 (2), pp 213-231, 1988.

Park S., Ramirez F., *Optimal Production of secreted protein in Fed-batch reactors*, AIChE Journal, 34 (9), pp 1550-1558, 1988.

Paynter H., *Analysis and design of engineering systems*, MIT press, 1961.

Perret J., *Intégration des réseaux de pétri différentiels à objets dans une plate-forme de simulation dynamique hybride : application aux procédés industriels*, Thèse de Doctorat INP Toulouse, France, 2003.

Petzold L., Li S., Cao Y., Serban R., *Sensitivity analysis of differential-algebraic equations and partial differential equations*, Computers and Chemical Engineering, 30, pp 1553-1559, 2006.

Petzold L., Serban R., Li S., Raha S., Cao Y., *Sensitivity Analysis and design optimization of differential algebraic equation systems*, Proceeding NATO advanced research workshop on computational aspects of non linear structural systems with large rigid body motion, 2000.

Pontryagin L., Boltyanski V., Gamkrelidze R., Michtchenko E., *Théorie Mathématique des processus optimaux*, Editions Mir, Moscou, 1974.

R

Raghunathan A., Diaz S., Biegler L., *An MPEC formulation for dynamic optimization of distillation operations*, Computers and Chemical Engineering, 28, pp 2037-2052, 2004.

Robinson E., *The optimisation of batch distillation operations*, Chemical Engineering Science, 24, pp 1661-1668, 1969.

Rosen O., Luus R., *Evaluation of gradients for piecewise constant optimal control*, Computers and Chemical Engineering, 15 (4), pp 273-281, 1991.

Ruppen D., Benthack C., Bonvin D., *Optimization of batch reactor operation under parametric uncertainty – computational aspects*, Journal of Process Control 5 (4), pp 235-240, 1995.

S

- Sargent R.W.H., Sullivan G.R., *Development of feed changeover policies for refinery distillation units*, Industrial & Engineering Chemistry Process Design and Development, 18 (1), pp 113-124, 1979.
- Sargousse A., *Noyau numérique orienté objet dédié à la simulation des systèmes dynamiques Hybrides*, Thèse de Doctorat INP Toulouse, France, 1999.
- Serban R., Petzold L.R., *COOPT - A Software Package for Optimal Control of Large-Scale Differential-Algebraic Equation Systems*, Journal Mathematics and Computers in Simulation 56 (2), pp. 187-204, 2001.
- Siebenthal C., Aris R., *Studies in optimization – VII. The application of Pontryagin's methods to the control of batch and tubular reactors*, Chemical Engineering Science, 19, pp.747-761, 1964.
- Simon A.M., Doran P., Paterson R., *Assessment of diffusion coupling effects in membrane separation. Part I Network thermodynamics modeling*, Journal of Membrane Science, 109, pp 231-246, 1996.
- Sorensen E., Macchietto S., Stuart G., Skogestad S., *Optimal control and on-line operation of reactive batch distillation*, Computers and Chemical Engineering, 20 (12), pp 1491-1498, 1996.
- Srinivasan B., Palanki S., Bonvin D., *Dynamic optimization of batch processes I. Characterization of the nominal solution*, Computers and Chemical Engineering, 27, pp 1-26, 2003.
- Stoer J., Bulirsch R., *Introduction to numerical analysis*, Springer Verlag, Berlin, 1980.

T

- Tanartkit P., Biegler L., *Stable Decomposition for dynamic optimization*, Industrial & Engineering Chemistry Research, 34, pp 1253-1266, 1995.
- Tanartkit P., Biegler L., *A nested, simultaneous approach for dynamic optimization problem I*, Computers and Chemical Engineering, 20 (6-7), pp 735-741, 1996.
- Tanartkit P., Biegler L., *A nested, simultaneous approach for dynamic optimization problem II: the outer problem*, Computers and Chemical Engineering, 21 (12), pp 1365-1388, 1997.
- Trélat E., *Contrôle Optimal Théorie et Applications*, Editions Vuibert, 2005.

V

- Valentin-Roubinet C., *Modelling of hybrid systems: DAE supervised by Petri nets, the example of a gas storage*, Automation of mixed Processes (ADPM98), pp 142-149, Reims France, 1998.
- Valentin-Roubinet C., *Hybrid Systems modeling : mixed Petri nets*, 3rd IMACS/IEEE conference CSCC99, pp 223-228, Athènes Grèce, 1999.
- Vanderbei R., Shanno D., *An interior point algorithm for nonconvex nonlinear programming*, Computational Optimization and Applications, 13, pp 231-252, 1999.
- Vassiliadis V.S., Sargent R.W.H., Pantelides C.C., *Solution of a class of multistage dynamic optimization problems. I Problems without path constraints*, Industrial & Engineering Chemistry Research, 33, pp 2111-2122, 1994 a.

Vassiliadis V.S., Sargent R.W.H., Pantelides C.C., *Solution of a class of multistage dynamic optimization problems. 2 Problems with path constraints*, Industrial & Engineering Chemistry Research, 33, pp 2123-2133, 1994 b.

Villiermaux J., *Génie de la réaction chimique, conception et fonctionnement des réacteurs*, TEC&DOC, Lavoisier, 1993.

Z

Zaytoon J., *Systèmes Dynamiques hybrides*, Hermes Sciences publications, 2001.