

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

MINISTRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

**UNIVERSITE ZIANE ACHOUR DE DJELFA**

Faculté des sciences et de la technologie



# Mémoire

Présenté pour l'obtention du diplôme de

**Magister en Génie électrique**

Option : **Automatique Avancée**

Par

**BELAZREG MOHAMED FOUZI**

Ingénieur d'État de l'ENP

**Intitulé**

**Commande des systèmes non linéaires par une approche  
hybride. Application au système « trois cuves »  
sous l'environnement APROS**

Soutenu le 09 avril 2015 devant le jury composé de :

<b>K. GUESMI</b>	<b>Professeur à U. de Djelfa</b>	<b>Président</b>
<b>D. BOUKHETALA</b>	<b>Professeur à l'E.N.P d'Alger</b>	<b>Rapporteur</b>
<b>K. HALBAOUI</b>	<b>Maitre de Recherche B au CRNB</b>	<b>Rapporteur</b>
<b>M. TADJINE</b>	<b>Professeur à l'E.N.P d'Alger</b>	<b>Examineur</b>
<b>S. SAADI</b>	<b>Maitre de Conférences A à U. de Djelfa</b>	<b>Examineur</b>
<b>M. S. BOUCHERIT</b>	<b>Professeur à l'E.N.P d'Alger</b>	<b>Invité</b>

2014/2015

# Remerciements

Ce travail présenté dans ce mémoire a été réalisé au sein du laboratoire de modélisation et simulation, département d'Automatique au Centre de Recherche Nucléaire de Birine 'CRNB' en collaboration avec l'Université Ziane Achour de Djelfa.

Je tiens tout d'abord à exprimer ma profonde gratitude à Messieurs DJAMEL BOUKHETALA, Professeur à l'École Nationale Polytechnique, et Dr. HALBAOUI KHALED, Maître de Recherche B au Centre de Recherche Nucléaire de Birine, respectivement directeur et co-directeur de mémoire, qui leur revient un grand mérite aussi bien pour la proposition du sujet et l'évolution et la valorisation du présent travail ainsi qu'à l'orientation et leurs encouragements durant l'achèvement du mémoire.

Je tiens aussi à remercier vivement le professeur GUESMI KAMEL et responsable de la filière de magister en automatique avancée à l'université de Djelfa, qui m'a fait l'honneur de présider mon jury de mémoire.

J'exprime aussi ma profonde gratitude à messieurs M.S. BOUCHERIT et M. TADJINE Professeurs à l'École Nationale Polytechnique, qu'ils n'ont jamais hésité à rester disponible pour nous orienter et nous conseiller pendant et après la formation.

Je souhaite remercier aussi Monsieur S. SAADI Maître de Conférences A à l'université de Djelfa, d'avoir accepté d'examiner ce mémoire.

Je souhaite aussi remercier Monsieur M.H. BELHEOUCHAT, Chargé de Recherche au Centre Nucléaire de Birine, qui n'a jamais hésité de m'apporter son aide et ses encouragements.

Mes remerciements vont également à Mr. A. KERRIS, Mr. BENAAZA, I. ABDELANI et S. Y. BENHACINE, respectivement Directeur Général, Secrétaire Général, Directeur de la Division Étude et Développement de l'Instrumentation Nucléaire et Chef du Département d'Automatique du Centre de Recherche Nucléaire de Birine.

Je remercie aussi mes collègues de travail H. AKROUM, H.D BABAAMEUR, A.H AZZOUZ, AER. BERRAHAL, Y. BOUACHAOUI, AOUICHAT et R. BENABID pour leurs soutiens et encouragements pendant le parcours de deux années d'étude et formation en magister.

Un remerciement très spécial à ma femme qui a bien su comment m'aider à mener à terme ce travail.

Enfin, je remercie toute les personnes qui m'ont aidé de près ou de loin à finaliser ce mémoire.

## DÉDICACE

*Je dédie ce modeste mémoire*

*A la mémoire de mes parents,*

*A ma femme pour sa précieuse aide*

*A mes filles Yasmine, Aicha Dorsaf et Fatima*

*Zahra*

*A mes frères et mes sœurs*

**ملخص-** تعرض هذه المذكرة النمذجة والتحكم في النظم الهجينة باستخدام إطار نظام مختلط منطقي و ديناميكي، معرف بقوانين فيزيائية مترابطة ، وقواعد منطقية، إضافة إلى قيود التشغيل. يتم وصف هذه الأخيرة بمعادلات ديناميكية خطية تخضع للمترجمات الخطية التي تستخدم متغيرات حقيقية و صحيحة. و باستخدام المتغيرات الملحقة، يمكن نمذجة التغيرات التي قد تظهر على هذه الديناميكيات مع الأخذ بعين الاعتبار الترابط الموجود بينها.و يتم استخدام هذا النموذج لتخليق قانون تحكم تنبؤي للنظام الهجين. هذا التحكم يستخدم حل أي باستعمال برنامج مزدوج صحيح تربيعي أو خطي لحل مشكلة الإستمثال.رغم ما تمثله هذه الطريقة (MIQPs) من محاسن إلا أن كثافة الحساب فيها ينجم عنها إستغراق زمن طويل، مما يتطلب إستعمال برامج فعالة او اللجوء إلى طريقة غير آتية في مشكلة الإستمثال. من أجل إظهار نجاعة و كفاءة هذه الطرق و الأدوات، تم إجراء المحاكاة باستخدام برنامج HYSDEL، البرمجة متعددة الوسائط MPT و برنامج APROS مطبقة على نظام ثلاث خزانات.

**الكلمات المفتاحية:** الأنظمة الهجينة، نظام مختلط منطقي و ديناميكي، نموذج التحكم التنبؤي، البرمجة المزدوجة الصحيحة التربيعية أو الخطية، البرمجة المتعددة الوسائط، برنامج APROS.

*Abstract*—This work deals with the modeling and the control of hybrid systems by using Mixed Logical Dynamical (MLD) system framework described by interdependent physical laws, logic rules, and operating constraints. These are described by linear dynamic equations subject to linear inequalities involving real and integer variables. The changes which may appear over such dynamics are modeled by using the auxiliary variables which take into account the interconnections effects. The MLD model is used to synthesize the model predictive control (MPC) law. The discrete-time equivalent of the model predicts the hybrid system behavior over a prediction horizon. The controller requires solution of on line mixed integer quadratic or linear program to solve an optimization problem. However, an explicit form is permitted in off-line optimization, in order to reduce the computation load due to the complexity of the MIQPs to be solved in reel time. Thus, the on-line control computation reduces to the simple evaluation of an explicitly defined piecewise linear function. Simulation was performed using HYSDEL compiler, MPT (Multiparametric Programming Toolbox), and APROS software to illustrate performances and efficiency of these tools using the model of a three-tank COSY benchmark.

**Keywords:** Hybrid system, non-linear systems, Mixed logical and dynamical, model predictive control, Mixed integer quadratic/linear programming, APROS, Multi-Parametric Programming.

**Résumé** - Ce mémoire présente la modélisation et la commande des systèmes hybrides en utilisant le formalisme dynamique logique mixte (MLD) décrit par des lois physiques interdépendantes, règles logiques et des contraintes d'exploitation. Ils sont régis par des équations dynamiques linéaires soumis à des inégalités linéaires faisant intervenir des variables réelles et entières. Les changements qui peuvent apparaître sur ces dynamiques sont modélisés en utilisant les variables auxiliaires prenant en compte les interconnexions. Le modèle MLD est utilisé pour la synthèse d'une commande prédictive (MPC). Le système à temps discret équivalent du modèle prédit le comportement dynamique du système hybride sur un horizon de prédiction. La commande nécessite une solution en ligne en utilisant un programme mixte entière quadratique ou linéaire pour résoudre un problème d'optimisation. Cependant, une forme explicite est permise dans l'optimisation off-line, afin de réduire le volume de calcul en raison de la complexité des MIQPs à résoudre en temps réel. Ainsi, le calcul de la commande en ligne est réduit à une simple évaluation d'une fonction linéaire par morceaux explicitement définie. La simulation a été effectuée en utilisant le compilateur HYSDEL, MPT (boîte à outils de la programmation Multi-Paramétrique), et le logiciel APROS afin d'évaluer les performances et l'efficacité de ces outils en utilisant le modèle d'un benchmark COSY à trois réservoirs.

**Mots clés :** Systèmes hybrides, systèmes non-linéaires, Système MLD, la commande prédictive à horizon glissant, la programmation linéaire entière quadratique, APROS, La programmation linéaire multi-paramétriques.

## Table des matières

Remerciements.....	2
Dédicace.....	3
Résumé.....	4
Liste des figures et tableaux.....	7
Abréviation.....	9
<b>Introduction Générale.....</b>	<b>11</b>

### **Chapitre I : Les systèmes dynamiques hybrides : Concepts Théoriques**

I.1 Introduction.....	15
I.2 Notions de système hybride.....	16
I.2.1 Structure d'un système dynamique hybride.....	17
I.2.2.1 La partie discrète.....	17
I.2.2.2 La partie continue.....	18
I.2.2.3 Interface entre la partie continue et la partie discrète.....	18
I.3 Les applications pratiques des systèmes hybrides.....	19
I.4 Catégories de phénomènes hybrides.....	19
I.4.1 Définition formelle des SDH.....	20
I.4.2 Classifications des phénomènes hybrides.....	21
I.5 Autres phénomènes considérés comme système hybride.....	23
I.6 Exemple de classes de systèmes dynamiques hybrides.....	23
I.7 Modélisation des systèmes dynamique hybrides.....	27
I.7.1 Automate hybride discret (DHA).....	27
I.7.2 Modèle à complémentarité linéaire.....	28
I.7.3 Modèle à complémentarité linéaire étendue ELC.....	28
I.7.4 Modèle Max-Min-Plus-Scaling MMPS.....	29
I.7.5 Modèle dynamique et logique MLD.....	30
I.7.6 Modèle affine par morceaux PWA.....	30
I.8 Équivalence entre les modèles les modèles.....	32
I.9 Outils de simulation, langages de programmation et logiciels pour les systèmes hybrides.....	32
I.9.1 Outils issus des systèmes continus étendus aux systèmes dynamiques hybrides.....	33
I.9.2 Outils issus des systèmes discrets étendus aux systèmes dynamiques hybrides.....	34
I.9.3 Outils pour les systèmes dynamiques hybrides (langage CT/CE).....	34
I.9.4 Outils de simulation.....	34
I.10 Conclusion.....	35

### **Chapitre II : Modélisation & Simulation du système hybride trois-cuves sous APROS**

II.1 Introduction.....	37
II.2 Architecture modulaire d'APROS.....	37
II.2.1 Architecture dynamique et numérique d'APROS.....	38
II.2.2 Modèle hydrodynamique.....	39
II.2.2.1 Le régime stationnaire.....	40
II.2.2.2 Le régime transitoire.....	40
II.3 Description de l'environnement de modélisation Grades.....	40
II.3.1 Structure hiérarchique de la modélisation.....	40
II.3.2 Organisation interne de modèle de procédé.....	41

# Table des matières

---

II.3.3 Les mécanismes de communications avec les applications externes.....	42
II.4 Modélisation des circuits hydrauliques.....	43
a. Circuit hydraulique avec un contrôleur PI de débit.....	43
b. Circuit avec prédicteur de Smith pour le contrôle de la température.....	44
II.5 Description du benchmark COSY « trois cuves ».....	45
II.5.1 Modélisation mathématique du système à trois cuves.....	46
II.5.2 Modélisation du système à trois cuves sous l'environnement APROS.....	47
II.6 Conclusion.....	49

## Chapitre III : Commande prédictive d'un système hybride sous forme MLD

III.1 Introduction.....	51
III.2 Description du formalisme de modélisation MLD des systèmes hybrides.....	51
III.2.1 Calcul proportionnel.....	52
III.2.2 Modélisation mathématique par le formalisme MLD.....	53
III.2.2.1 Architecture du formalisme MLD.....	53
III.2.2.2 Mise en équation du formalisme MLD.....	54
III.3 La commande prédictive.....	55
III.3.1 Les idées principales de la commande prédictive.....	55
III.3.2 Modèle de prédiction pour la commande MPC.....	57
III.4 La commande prédictive d'un système hybride sous forme MLD.....	58
III.4.1 Structure du critère d'optimisation.....	58
III.4.2 Formulation du critère d'optimisation pour la méthode de Bemporad.....	62
III.4.3 Programmation de l'algorithme d'optimisation quadratique MIQP.....	64
III.4.4 Description de la méthode de résolution Branch&Bound « B&B ».....	65
III.5 La commande prédictive explicite hors-ligne.....	69
III.6 Commande prédictive des systèmes hybrides par une approche MLD multi-modèle.....	71
III.6.1 Domaines caractéristiques.....	72
III.6.2 Développement de la stratégie de commande.....	72
III.6.3 Découpage en sous-régions pour des modèles multiples.....	73
III.7 Conclusion.....	75

## Chapitre IV : Application et Simulation sur un système hybride à trois cuves

IV.1 Introduction.....	77
IV.2 Description du banc d'essai « trois cuves » sous l'environnement APROS.....	77
IV.3 Mise sous forme MLD.....	78
IV.4 Simulation & Interprétation.....	79
IV.5 Conclusion.....	91

<b>Conclusion générale et Perspectives.....</b>	<b>94</b>
---	-----------

<b>Annexes.....</b>	<b>95</b>
---------------------	-----------

<b>Références bibliographiques.....</b>	<b>115</b>
---	------------

<b>Production scientifique dans le cadre des travaux de Magister.....</b>	<b>119</b>
---	------------

## Liste des figures et tableaux

Figure I.1	Couplage de phénomènes continus et discrets au sein d'un système dynamique hybride.....	16
Figure I.2	Système hybride avec 4 modes de fonctionnement.....	17
Figure I.3	Structure d'un système dynamique hybride.....	17
Figure I.4	Phénomènes hybrides.....	22
Figure I.5	Automate hybride de la basse à rebondissement.....	24
Figure I.6	Système hybride du thermostat.....	25
Figure I.7	Évolution de la température de la chambre.....	25
Figure I.8.a	Modèle d'un système hydraulique.....	26
Figure I.8.b	Modèle d'une vanne pneumatique.....	26
Figure I.9	Système composé par deux chariots.....	28
Figure I.10	Équivalence entre les différents modèles hybrides.....	32
Figure II.1	Organisation hiérarchique de l'outil de modélisation & Simulation APROS.....	37
Figure II.2	Structure modulaire des calculs transitoires dans APROS.....	38
Figure II.3	Architecture différentielle et algébrique du code APROS.....	38
Figure II.4	Méthode de résolution numérique adoptée par APROS.....	39
Figure II.5	Structure hiérarchique du modèle APROS.....	41
Figure II.6	Organisation hiérarchique du projet APROS.....	42
Figure II.7	Les Interfaces de communications externes supportées par APROS.....	42
Figure II.8	Modèle de procédé hydraulique APROS.....	43
Figure II.9	Simulation du transitoire avec contrôle de débit du procédé hydraulique.....	43
Figure II.10	Processus thermique avec régulateur de température par le prédicteur de Smith.....	44
Figure II.11	Simulation du contrôle de température en utilisant un prédicteur de Smith.....	45
Figure II.13	Banc d'essai à trois cuves COSY.....	45
Figure II.14	Modélisation du système à trois cuves par logiciel APROS.....	47
Figure II.15	Simulation en boucle ouverte du système à trois cuves sous APROS.....	48
Figure III.1	Structure d'un modèle MLD.....	54
Figure III.2	Principes de la commande prédictive.....	56
Figure III.3	Schéma caractérisant la commande prédictive.....	57
Figure. III.4	Stratégie de commande MPC hybride en ligne appliquée au système à trois-cuves sous l'environnement APROS.....	65
Figure. III.5	Arbre binaire d'un problème MIQP à trois variables entières (stratégie de type « premier en profondeur).....	67
Figure III.6	Arbre binaire d'un problème <i>MIQP</i> à trois variables entières. Les nombres correspondent à la stratégie de résolution (ici stratégie de type « premier selon la largeur ».....	68
Figure III.7	Stratégie de la commande hybride hors-ligne appliquée au système à trois-cuves sous APROS.....	71
Figure III.8	Partition de l'espace d'état.....	72
Figure III.9	Stratégie de commande Multi-modèles MPC hybride en ligne appliquée au système à trois-cuves sous l'environnement APROS.....	74
Figure IV.1	Modèle APROS du système à trois-cuves « COSY ».....	78
Figure IV.2	Commande prédictive des niveaux $h_1$ , $h_2$ et $h_3$ pour $N=3$ et $V_{L3}=1$ .....	79
Figure IV.3	Commande prédictive des niveaux $h_1$ , $h_2$ et $h_3$ pour un état de blocage $V_{L3}$ .....	81
Figure IV.4	Commande prédictive des niveaux $h_1$ , $h_2$ et $h_3$ , $V_{L3}$ ouverte avec variation de consigne.....	82
Figure IV.5	Commande prédictive des niveaux $h_1$ , $h_2$ et $h_3$ avec contrôle sur la vanne $V_{L3}$ .....	83
Figure IV.6	Commande prédictive des niveaux $h_1$ , $h_2$ et $h_3$ , $V_{L3}=1$ avec consigne en forme sinusoïdale.....	84
Figure IV.7	Commande prédictive des niveaux $h_1$ , $h_2$ et $h_3$ , $V_{L3}=1$ avec des consignes variables.....	85
Figure IV.8	Commande prédictive des niveaux $h_1$ , $h_2$ et $h_3$ , $V_{L3}$ contrôlée avec variation des consignes sur les trois niveaux de liquide.....	86
Figure IV.9	Commande prédictive des niveaux $h_1$ , $h_2$ et $h_3$ , $V_{L1}$ , $V_{L3}$ contrôlées avec variation de consignes sur les trois niveaux de liquide.....	88

## Liste des figures et tableaux

---

Figure IV.10	Commande prédictive des niveaux $h_1$ , $h_2$ et $h_3$ , $V_{L1}$ , $V_{L2}$ , $V_{L3}$ contrôlées avec variation de consignes sur les trois niveaux de liquide.....	89
Figure IV.11	Commande prédictive multi-modèle des niveaux $h_1$ , $h_2$ et $h_3$ avec $V_{L3}=1$ .....	90
Figure IV.12	Commande prédictive affine par morceaux PWA des niveaux $h_1$ , $h_2$ et $h_3$ (Horizon de prédiction $N=3$ , $Q_y=103$ , $Q_u=10-3$ , vannes VL3 est maintenue ouverte avec variation en échelon des consignes des niveaux $h_1$ , $h_2$ et $h_3$ ).....	91

## Liste des tableaux

Tableau I.1	Déclenchement des phénomènes hybrides.....	21
Tableau I.2	Actions des phénomènes hybrides.....	21
Tableau II.1	Paramètres du benchmark trois-cuves "Three-tank".....	46
Tableau III.1	Représentation compacte des tables de recherche contenant la structure de la commande.....	71



# Abréviations

APROS	Advanced Process Simulation
ATWS	Anticipated transient without scram
BaSiP	Batch Simulation Package
CARIMA	Controlled Auto-Regressive Integrated Moving Average
COSY	Control of Complex Systems
GPC	Generalized Predictive Control
gPROMS	General PROcess Modeling System
HSCAP	hybrid Sequential Causal Assignment Procedure
HSML	Hybrid System Modeling Language
HyBrsim	hybrid Bond graph Simulator
HYSDEL	Hybrid system Description Language
LOCA	Loss of Cooleant Accident
LOFA	Loss Of Flow Accident
MILP	Mixed Integer Linear Programming
MIQP	Mixed Integer Quadratic Programming
MPC	Model Predictive Control
MPT	Multi-Paramatric Toolbox
Omola /Omsim	Object-Oriented Modeling Language
OPC	OLE for Control Process
RELAP	Reactor Excursion and Leak Analysis Program
RdPH	Réseau de Petri hybride
RHC	Receding Horizon Control
SED	Système à événement discret
YAHMST	Yet Another Hybrid Simulation Tool
YALMIP	Yet Another LMI Parser
VTT	Technical Research Centre of Finland

# *Introduction Générale*

# *Introduction générale*

L'appellation « systèmes dynamiques hybrides » désigne des catégories de systèmes faisant intervenir explicitement simultanément des phénomènes ou des modèles de type dynamique continue et de type événementiel. Ces systèmes sont classiquement constitués de processus continus interagissant avec ou supervisés par des processus discrets. Ils résultent également de l'organisation hiérarchique de systèmes de contrôle/commande complexes ou de l'interaction entre des algorithmes discrets de planification et des algorithmes continus de commande [1][2].

En automatique classique, le problème de commande est traité différemment selon la nature du système à étudier : continu ou discret. L'étude et l'analyse du système est basée sur un ensemble de théories et concepts faisant aboutir à des méthodes et développement de solutions performantes permettant le réglage du problème dans sa nature homogène sans faire appel aux solutions et apport de l'autre domaine. Une théorie générale et unifiée pour modéliser, analyser et concevoir des commandes pour de tels systèmes n'est pas disponible actuellement. Cependant, beaucoup de chercheurs travaillent sur des sous-classes de systèmes hybrides pour lesquelles de telles approches ont été développées [1][3][4].

Sachant que les procédés industriels sont de nature complexes et sont difficiles à commander, et pour les piloter les ingénieurs utilisent des systèmes de contrôle et commande par des automates séquentiels couplés avec des boucle de régulation dans les différents modes de fonctionnement du système. Cet aspect de fonctionnement dynamique est caractérisé par une nature continue et événementielle.

A cet effet, l'étude des systèmes hybrides a retenu l'attention de la communauté des automaticiens, ainsi que celle de la communauté des informaticiens. Les objectifs que l'on peut assigner à l'étude des systèmes dynamiques hybrides SDH consistent à apporter une solution en termes de modèle, de méthode, de performance et de qualité à des problèmes mal traités par les approches homogènes ou classiques. En effet, la communauté des chercheurs a orienté ces efforts vers trois axes : modélisation, analyse et commande [2].

La modélisation permet de formaliser des modèles pour décrire le comportement dynamique des SDH. Plusieurs formalismes ont été proposés afin d'établir un modèle homogène permettant l'interaction entre la partie continue et discrète. Ces approches peuvent être classées en trois catégories [5]:

1. Les approches basées sur une extension des modèles continus. Parmi elles, une extension des modèles bond-graph,
2. Les approches basées sur une extension des modèles pour les systèmes à événements discrets, parmi elles les réseaux de Pétri RdP hybrides,

3. Les approches mixtes combinant les parties continues et discrètes dans une même structure. Ces modèles reposent sur la collaboration de deux sous modèles. L'un pour l'aspect événementiel, et l'autre pour l'aspect continu basé sur des équations d'état dont l'aspect hybride est pris en compte dans l'interface entre les deux sous-modèles.

Différents formalismes sont proposés pour modéliser les systèmes hybrides. Les plus connus sont les automates linéaires [6], les systèmes affines par morceaux 'PWA' [7], les systèmes à complémentarité linéaire 'LC' [8], les systèmes à complémentarité linéaire étendue 'ELC' [9], les systèmes Max-Min-Plus Scaling 'NMPS' [10], les systèmes mixtes dynamiques et logiques 'MLD' [3].

Les approches développées dans ce mémoire adoptent ce dernier formalisme MLD, proposé par Bemporad [3]. Il permet une modélisation des systèmes hybrides, incluant des dynamiques continues et discrètes, leurs interactions, ainsi que des contraintes. Par ailleurs, la commande prédictive à base de modèle (MPC) qui est largement diffusée dans l'industrie comme technique de commande simple et efficace est utilisée pour commander un système hybride sous forme MLD en termes de poursuite de trajectoire. La structure de commande prédictive à horizon glissant développée sous le formalisme MLD est basée sur une technique d'optimisation quadratique mixte entière.

C'est dans ce cadre que s'inscrit notre sujet de mémoire de magister et qui a pour objectif d'étudier une méthode de modélisation en vue de commander un système non-linéaire par une approche hybride. Notre contribution réside dans l'utilisation d'un outil de simulation APROS (Advanced Process Simulation) destiné pour la simulation et l'analyse de fonctionnement des centrales nucléaires et thermiques d'une part, et le développement mathématique de l'algorithme de commande hybride sur les deux aspects prédiction et optimisation d'autre part

Le travail de recherche réalisé dans le cadre de ce mémoire est résumé en trois points :

1. Etude et simulation des systèmes dynamiques hybrides par l'utilisation du logiciel APROS,
2. Etude et application d'une commande basée sur une stratégie prédictive à base de modèle (MPC) sur un système hybride sous forme MLD et multi MLD,
3. Etude et application d'une commande basée sur une approche prédictive explicite **piecewise** pour le calcul de la loi de commande qui se traduit par une table **lookup** En-ligne en fonction du vecteur d'état.

Les deux dernières décennies ont vus croître l'intérêt porté à la recherche concernant les systèmes hybrides car ils couvrent plusieurs domaines d'application tels que les processus batch, les systèmes électrotechniques, les réseaux de communication, les systèmes de transport, la commande de processus industriels, les systèmes flexibles de production, la robotique, ... etc.

Les progrès rapides de l'informatique et de la technologie des ordinateurs encouragent l'analyse des systèmes hybrides car les ordinateurs sont capables de résoudre des calculs de plus en plus complexes.

Le formalisme **MLD** s'adapte à de très nombreuses classes de systèmes hybrides, par exemple, les systèmes à entrées et états mixtes (continus et discrets). Il permet de résoudre les problèmes pratiques tels que la commande et l'estimation des états. La structure **MLD** fournit des équations dynamiques linéaires à temps discret, avec des contraintes exprimées par des inégalités, impliquant des variables réelles et entières. Elle s'intègre dès lors facilement dans la formulation standard de problèmes d'estimation ou de commande.

Dans le premier chapitre, les notions fondamentales relatives aux systèmes dynamiques hybrides sont introduites. Les caractéristiques, les différentes classes et phénomènes hybrides cités dans la littérature sont présentés. Ensuite, une description succincte est donnée sur les principaux formalismes de modélisation des systèmes hybrides.

Dans le deuxième chapitre nous nous sommes intéressés à la modélisation et la simulation sous l'environnement du logiciel **APROS** qui permet d'établir un modèle thermo-hydraulique, à partir de composants de bases liées entre eux par des lignes et des jonctions, constituant ainsi un modèle qui peut s'étendre pour un cas plus complexe. La simulation et le paramétrage des différents composants est basé sur des boîtes de propriétés où seront introduites les caractéristiques physiques tels que : la longueur, la surface, type de fluide, la rugosité,...etc. La méthode de modélisation adoptée par le logiciel est appelée la **nodalisation** ou le découpage est appliquée sur le système à trois cuves COSY.

La démarche de modélisation d'un système hybride avec le langage **HYSDEL** sous le formalisme **MLD** est présentée au troisième chapitre. Ensuite, une description est donnée sur les stratégies de commande prédictive à base de modèle (**MPC**) à horizon glissant pour un système hybride modélisé sous forme **MLD**, **MLD** multiple, incluant plusieurs modèles, ainsi que la stratégie *piecewise* (**PWA**) explicite en utilisant la bibliothèque **MPT** (**MultiParamétrique Toolbox**). Enfin, le principe de la technique de résolution du problème d'optimisation **MIQP** « Branch & Bound » (**B&B**) est abordé.

Le quatrième chapitre est dédié à la simulation des techniques de commande développées au troisième chapitre et appliquées sur un modèle à trois cuves sous l'environnement **APROS** en utilisant l'interface **OPC**. Plusieurs configurations sont testées afin d'évaluer l'efficacité et la robustesse de la commande prédictive (**MPC**) pour un système hybride.

Enfin, nous terminerons notre travail par une proposition de perspectives.

# *Chapitre I*

Les Systèmes Dynamiques

Hybrides

Concepts théoriques

## I.1 Introduction

Le domaine des systèmes dynamiques hybrides retient de plus en plus l'intérêt des communautés scientifiques de l'automatique et de l'informatique, avec une augmentation de la production scientifique. Cette croissance est la conséquence directe des progrès considérables dans le domaine des moyens de calcul, les logiciels de simulation, des technologies de communication, des méthodes d'acquisition de données et d'actionneurs plus performants [11].

Cette classe de systèmes prenant explicitement en considération des phénomènes continus et discrets ainsi que les interactions pouvant en résulter au sein d'une structure commune, le champ des systèmes hybrides présente de fait de nombreux défis théoriques, mais est susceptible d'avoir un impact important sur les applications pratiques. L'intérêt croissant pour ce type de systèmes peut être observé par exemple par le nombre croissant de sessions consacrées aux systèmes hybrides dans les congrès internationaux en Automatique, comme « **American Control Conference ACC** » ou « **Conference on Decision and Control CDC** ». De plus, de nombreuses conférences sont consacrées spécifiquement aux systèmes hybrides, comme le Workshop international « **Hybrid Systems : Computation and Control HSCC** » ou la conférence biennale « **Automation des Processus Mixtes ADPM** » devenue ADHS. Enfin, Plusieurs journaux internationaux ont publié des numéros spécifiques sur les systèmes hybrides, comme « **Automatica** », « **System and Control Letters** », « **Transactions on Automatic Control** », « **International Journal of Hybrid Systems** ».

La notion de système dynamique hybride **SDH** est apparue en premier temps dans l'article [12] en 1987 où l'auteur insiste sur la nécessité de développer une théorie mêlant des signaux continus et discrets. Cette nécessité trouve son origine dans de nombreux domaines :

- La simulation de procédés complexes composés de plusieurs unités et de modes opératoires continus,
- La conception de procédés 'Batch' qui traitent par une recette séquentielle une matière représentée sous forme continue,
- La supervision de procédés et la sûreté de fonctionnement de systèmes mutli-modèle,
- La mise au point de contrôleurs intelligent, c'est-à-dire de contrôleurs continus mais adaptant leur réponse suivant une logique discrète,
- L'ordonnancement et la gestion de tâches pour des systèmes temps réels,
- La modélisation du comportement dynamique des systèmes à temps de réponse contraint par leur environnement,
- Le contrôle du trafic aérien,
- La robotique,
- Etc.,

Ce thème de recherche rassemble une communauté scientifique d'horizon des plus variés : mathématiciens, automaticiens, informaticiens, ... etc. A cet effet, les besoins se situent bien au niveau de la modélisation, de la simulation, et de l'analyse de ces systèmes pour concevoir une approche de commande unifiée.

## I.2 Notion de système hybride :

Le terme hybride se réfère au couplage essentiel de phénomènes continus et discrets au sein d'un système comme est indiqué sur la figure I.1. Un système dynamique hybride permet de représenter globalement l'interdépendance d'éléments dynamiques continus/discrets au sens classique d'équations différentielles ou équations aux différences, soumis à des éléments décisionnels discrets déterministes ou systèmes à événements discrets **SED**. De telles représentations sont caractéristiques de diverses situations :

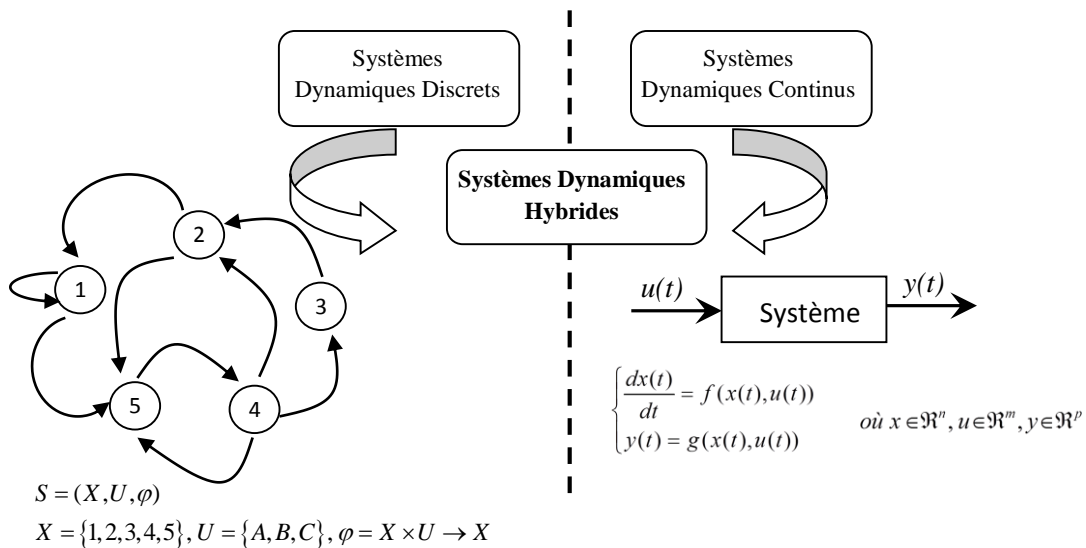


Figure I.1 : Couplage de phénomènes continus et discrets au sein d'un système dynamique hybride [1].

- ✓ Couplage entre un niveau décisionnel discret et des dynamiques continues,
- ✓ Couplage intervenant lors d'une synthèse par bouclage avec multi-régulateurs où le choix du régulateur dépend du comportement lié au point de fonctionnement du système,
- ✓ Couplage entre un système multi-modèle incluant des commutations entre ces modèles.

Les automates hybrides sont un modèle usuel de systèmes hybrides combinant des diagrammes de transition pour le comportement discret d'une part, et des équations différentielles pour les comportements continus d'autre part.

En principe, un système hybride présente un comportement caractérisé par plusieurs modes de fonctionnement. Dans chaque mode, l'évolution des états continus du système est décrite par une équation différentielle propre. Le système hybride commute entre les différents modes



lorsqu'un événement particulier se produit. Les liens représentés sur la figure I.2 sont utilisés pour indiquer les événements qui provoquent des changements entre les dynamiques du système hybride.

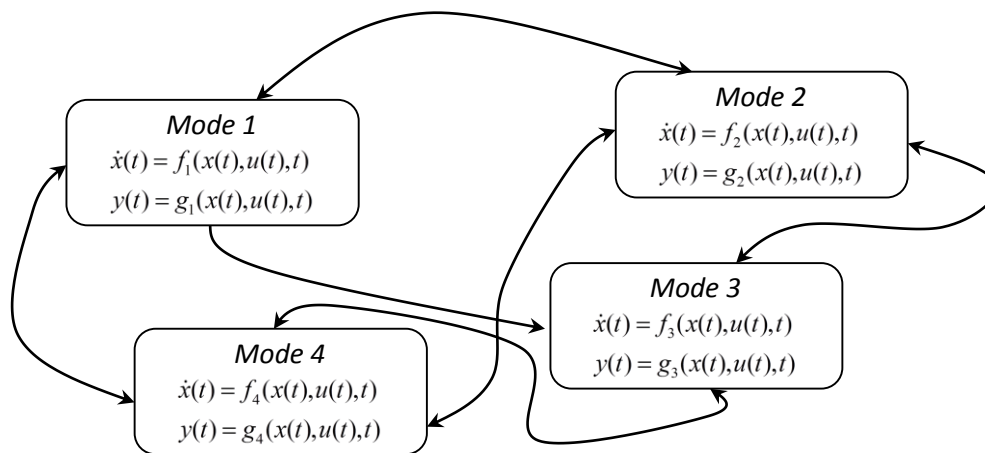


Figure. I.2 : Système hybride avec 4 modes de fonctionnement.

### I.2.1 Structure générique d'un système dynamique hybride :

Généralement, un SDH peut être vu comme l'agrégation d'un SED, de systèmes dynamiques continus et d'une interface qui gère les interactions entre les deux évolutions (continue et discrète) [13]. Chaque partie du SDH dispose d'entrées (discrète  $\sigma$ , continue  $u$ ) et de sortie (discrètes  $\psi$ , continues  $y$ ). L'état discret est décrit par  $q$  et l'état continu est décrit par  $x$ . Ces différents éléments, à savoir la partie discrète, la partie continue et l'interface peuvent être décrits ou spécifiés de différentes manières (figure I.3) :

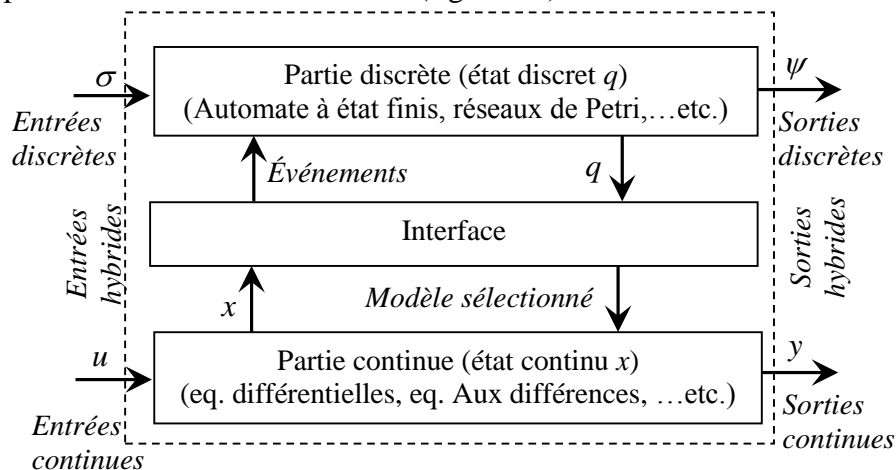


Figure I.3 : Structure d'un système dynamique hybride (Interaction entre la partie continue et la partie discrète)

#### I.2.1.1 La partie discrète :

La partie discrète du système hybride est associée à un SED dont l'espace d'état est un ensemble discret fini, où l'état discret ne change qu'à certains instants du temps de façon instantanée. Les transitions entre les états discrets sont associées à l'occurrence d'événements discrets. Les

modèles qui permettent d'appréhender les SED sont les automates à états finis, les réseaux de Pétri **RdP** et les graphes d'état (State Charts).

### I.2.1.2 La partie continue :

La partie continue est représentée par un ensemble de modèles continus. Elle est constituée d'une infinité de valeurs décrivant le comportement d'un phénomène physique. L'évolution de la partie continue est caractérisée par un espace de temps continu et peut être représentée de différentes manières. Les modèles souvent utilisés sont ceux à base d'équations différentielles ordinaires, ou encore de fonction de transfert, ou bond-graphs. Le modèle de comportement continue est exprimé sous la forme d'état :

$$y(t) = \begin{cases} \dot{x}(t) = f(x(t), z(t)) \\ h(x(t), z(t)) \end{cases}$$

Où :  $x(t)$  est la vecteur d'état,  $y(t)$  est la vecteur des sorties,  $z(t)$  est un vecteur regroupant les entrées inconnues, les perturbations, etc. et  $f$  et  $h$  sont des fonctions vectorielles linéaires ou non linéaires.

### I.2.1.3 Interface entre la partie continue et la partie discrète :

L'interface traduit l'interaction entre la partie continue et la partie discrète du SDH de façon que l'état discret  $q$  détermine la dynamique continue spécifique de la partie continue du SDH [13]. Par ailleurs, les signaux provenant de la partie continue déterminent par l'intermédiaire de l'interface les événements autorisant des transitions de l'état discret d'une situation vers une autre situation. Ce phénomène d'interaction entre ces types de dynamiques est appelé **phénomène hybride**.

Les actions des phénomènes hybrides se profilent généralement sous forme d'une commutation de modèle continu ou sous forme d'une discontinuité apparaissant sur le vecteur d'état. Dans le cas de commutation de modèle, le choix de la nouvelle dynamique est déterminé par le système à événements discrets décrivant la dynamique discrète du système global. Par ailleurs, ces deux actions hybrides peuvent être couplées.

Les phénomènes hybrides peuvent être déclenchés lorsque des variables continues atteignent certains seuils spécifiques. Ce fait se manifeste en faisant basculer le système à événement discret à un nouvel état et suivant cet état une des actions décrites ci-dessous est déclenchée. Ce phénomène est dit alors autonome [13][14]. En revanche, une commande discrète extérieure agissant directement sur le système à événement discrets peut déclencher un phénomène hybride. Dans ce cas, le phénomène hybride est dit contrôlé. Ainsi, quatre types de phénomène hybrides peuvent être distingués selon le type de déclenchement et selon les actions des phénomènes hybrides. Ces phénomènes par référence aux types de déclenchement et aux actions sont appelés les commutations de modèle autonomes, les commutations de modèle contrôlées, les sauts de l'état autonomes et les sauts de l'état contrôlés. Un SED peut se caractériser par plusieurs types de phénomènes [15].

### I.3 Les applications pratique des systèmes hybrides :

Grâce aux avancées des recherches et des développements dans les techniques de modélisation et d'analyse des systèmes hybrides, le nombre d'applications pratiques mettant en jeu des systèmes hybrides croît régulièrement. Plusieurs applications récentes sont citées dans la littérature comme exemple :

1. Complex gas supply system [3] : optimal control using MLD framework
2. Système à trois réservoirs [16] : fault modeling capability of MLD framework
3. Systèmes de gestion de trafic aérien [17],
4. La gestion de convois de véhicules sur autoroute [18],
5. Système Hydraulique [19] : Model predictive control using MLD model,
6. Système de production Multi-Batch [20] : Production optimale en utilisant MLD framework,
7. Evaporateur Batch [21] : Algorithm d'atteignabilité pour le framework MLD,
8. Process Hydro-Electrique [22]: Modélisation par le formalisme MLD,
9. Processus Batch de cristallisation de sucrerie [23] : modélisation et optimisation par MLD,
10. Contrôle des feux de circulation [24] : Optimal control of a class of linear hybrid systems with saturation

Les structures théoriques développées dans ce mémoire seront validées sur un modèle APROS d'un benchmark COSY incluant trois cuves, classiquement utilisées dans le domaine de la modélisation et la commande de systèmes hybrides.

### I.4 Catégories de phénomènes hybrides :

Parmi les premières études sur les systèmes hybrides, on trouve les systèmes à structure variable représentés par un ensemble fini de dynamiques continues et une commutation logique entre elles.

La première formulation unitaire des concepts concernant les systèmes hybrides a été réalisée par Branicky[15]. Les paragraphes ci-dessous précisent les principales définitions utiles à la compréhension, la classification et les modèles classiquement adoptés. A cet effet, on présente les actions discrètes qui peuvent intervenir lors de l'évolution d'un système continu décrit par un système différentiel de la forme :

$$\dot{x}(t) = \xi(t), \quad t \geq 0$$

Où :  $x(t) \in \mathcal{R}^n$  est le vecteur d'état du système. On note  $\tau$  l'instant où intervient une action discrète dont nous verrons ultérieurement comment elle peut être déclenchée. L'ensemble des actions et déclenchements est appelé phénomène hybride [15].

### Actions des phénomènes hybrides :

- Les phénomènes hybrides agissent sur la dynamique du système continu modifiant ainsi cette dynamique. Le système hybride se situe alors pour  $t \geq \tau$  dans un autre mode de

fonctionnement. On appelle ce phénomène **commutation/changement de modèle/dynamique** et  $\tau$  **instant de commutation**. Le choix du nouveau modèle est déterminé par le **SED**. Un exemple de modèle formel avec deux modes de fonctionnement est le suivant :

$$\dot{x}(t) = \xi_1(t) \quad t < \tau$$

$$\dot{x}(t) = \xi_2(t) \quad t \geq \tau$$

- Les phénomènes hybrides agissent également sur le vecteur d'état du système, le faisant évoluer de manière différente pour  $t = \tau$ . Une discontinuité apparaît alors sur le vecteur d'état. On appelle ce phénomène **saut de l'état**. Un exemple de modèle formel est représenté par les équations suivantes :

$$\dot{x}(t) = \xi(t) \quad t \geq 0 \text{ et } t \neq \tau$$

$$\dot{x}(\tau) = g(\cdot, \tau)$$

- Les deux actions peuvent être couplées ; en effet on peut considérer qu'à l'instant  $\tau$ , on a une commutation de modèle et un saut du vecteur d'état. un exemple de modèle formel est le suivant :

$$\dot{x}(t) = \xi_1(t) \quad t \leq \tau$$

$$\dot{x}(\tau) = g(\cdot, \tau)$$

$$\dot{x}(t) = \xi_2(t) \quad t > \tau$$

La fonction  $g$  provoque une discontinuité de l'état à l'instant  $\tau$ , puis intervient la deuxième dynamique continue  $\xi_2$ .

### Déclenchements des phénomènes hybrides :

- Les phénomènes hybrides peuvent être déclenchés lorsque le vecteur d'état continu atteint certaines valeurs. L'événement ainsi créé fait évoluer le SED celui-ci passe dans un nouvel état et suivant cet état une des actions décrites ci-dessous est générée sur le système continu. On dit alors que ce **phénomène est autonome**.
- Les phénomènes hybrides peuvent également être déclenchés par une commande discrète extérieure agissant sur le SED. On dit alors que **ce phénomène est contrôlé**.

#### I.4.1 Définition formelle des SDH :

Définition formelle des SDH :

On propose la définition formelle des systèmes dynamique suivante :

##### Définition :

Soit  $x(t) \in X \subset \mathcal{R}^n$ ,  $q(t) \in Q \subset N^m$ ,  $u(t) \in U \subset \mathcal{R}^c$ ,  $v(t) \in \Omega_c \subset N^d$  et  $t \in N^+$  où  $n, m, c, d$  sont données.

L'ensemble  $X$  représente l'ensemble des états continus et l'ensemble  $Q$  représente l'ensemble des états discrets. L'ensemble  $U$  représente l'ensemble des commandes continues et

l'ensemble  $\Omega_c$  représente l'ensemble des commandes discrètes. La variable  $t$  représente le temps.

Soit  $S = X \times Q$  et soit  $P$  un sous-ensemble fermé de  $R^n \times Q$ . On définit l'état hybride  $s(t) \in S$  du système à l'instant  $t$  par la donnée du couple  $[x(t), q(t)]$  et on appelle SDH un système dynamique décrit par les équations suivantes pour  $t \in [t_0, t_f]$  :

$$\dot{x}(t) = F(x(t), q(t), u(t)) \quad \forall s(t) \in S \cap C_p$$

$$[x(t^+), q(t^+)] = G(x(t), q(t), v(t)) \quad \begin{cases} \text{si } s(t) \in \partial P \\ \text{si } v(t) \in \Omega_c \end{cases}$$

$$\begin{cases} x(t_0) = x_0 \\ q(t_0) = q_0 \end{cases}$$

La fonction  $F$  représente la dynamique continue du système hybride et  $G$  représente la dynamique hybride du système correspondant aux phénomènes hybrides décrits précédemment. L'ensemble  $P$  est celui des zones de déclenchement des phénomènes hybrides et  $\partial P$  est la frontière de  $P$ . La commande discrète  $v(t)$  détermine les instants où la fonction  $G$  intervient (instants de commutation/saut). Elle détermine également le nouveau modèle ou l'amplitude du saut de l'état suivant l'action du phénomène hybride. Les fonctions  $F$  et  $G$  sont supposées suffisamment régulières pour que le système défini par les équations admette une solution unique.

#### I.4.2 Classification des phénomènes hybrides :

Dans cette partie, nous proposons une classification des phénomènes hybrides intervenant dans les SDH, sachant qu'un SDH peut posséder plusieurs types de phénomènes. Ces phénomènes étant décrits par la fonction  $G$ , on définit les différents phénomènes hybrides suivant les propriétés de cette fonction  $G$  et notamment suivant son ensemble de départ, noté  $D$  et son ensemble d'arrivée, noté  $A$ .

$D = X \times Q$	$D = \Omega_c$
Déclenchement Autonome	Déclenchement Contrôlé

Tableau I.1 : Déclenchement des phénomènes hybrides.

Le tableau 2 montre suivant l'ensemble d'arrivée de la fonction  $G$ , le type d'action correspondant.

$A = Q$	$A = X$
Commutation de modèle	Saut de l'état

Tableau I.2 : Actions des phénomènes hybrides.

Selon leur action et la manière dont ils sont déclenchés, on distingue quatre types de phénomènes hybrides [15].

- ✓ Les commutations de modèle Autonomes, notées  $G_{CA}$ .
- ✓ Les commutations de modèle contrôlées, notées  $G_{CC}$ .

- ✓ Les sauts de l'états Autonomes, notés  $G_{SA}$ .
- ✓ Les sauts de l'états contrôlés, notés  $G_{SC}$ .

Ces quatre phénomènes sont représentés graphiquement sur la figure I.4.

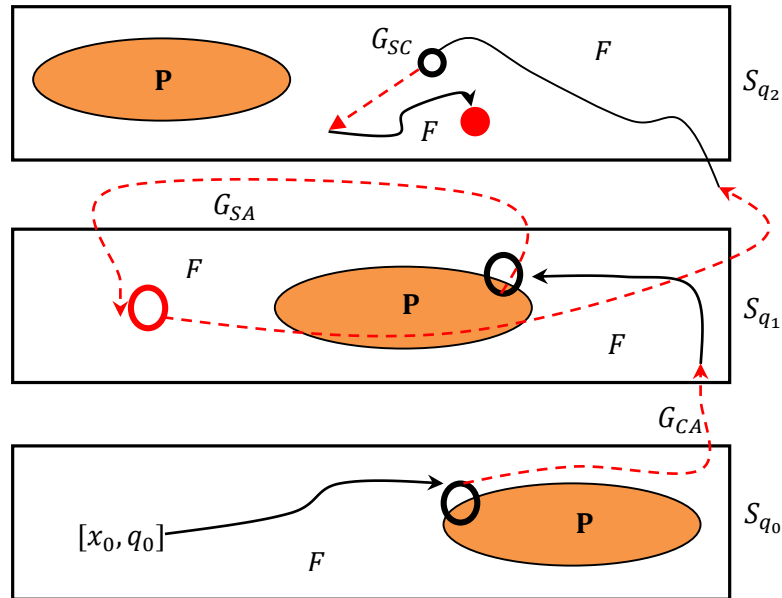


Figure I.4 : Phénomènes Hybrides.

Ainsi, en partant de l'état initial  $(t_0) = [x_0, q_0] \in S_{q_0}$ , le système évolue suivant la dynamique  $F$  jusqu'à ce que  $s$  atteigne l'ensemble  $P$ . A cet instant noté  $\tau_1$ , la dynamique hybride intervient et modifie la valeur de  $q$  qui devient égale alors à  $q_1$ . Ce phénomène est donc une commutation autonome ( $G_{CA}$ ).

Ensuite, le système poursuit son évolution avec une nouvelle dynamique définie par  $F$  (qui a été modifié). À l'instant  $\tau_2$ ,  $s$  atteint à nouveau l'ensemble  $P$ , mais cette fois la dynamique hybride modifie  $x$ . Ce phénomène est donc un saut autonome ( $GSA$ ).

Enfin, ces deux phénomènes se reproduisent mais sur l'intervention d'une commande discrète  $v(t)$  qui détermine dans un cas le nouveau modèle ( $GCC$ ) et dans l'autre cas la nouvelle valeur du vecteur d'état ( $GSC$ ).

A partir de ces considérations, Branicky a proposé la classification suivante concernant les catégories fondamentales des systèmes hybrides [15].

- Systèmes hybrides à commutation autonome (avec continuité de l'état) : caractérisées par un changement de  $\dot{x}$  lorsque l'état continu  $x$  atteint certains domaines dans l'espace des états continus, exemple d'un circuit électrique comportant une diode.

- Systèmes hybrides à saut autonome (avec discontinuité de l'état) : caractérisée par un changement discontinu de la variable d'état  $x$  lorsqu'elle atteint une certaine région de l'espace d'état, l'exemple de la balle en rebondissement.
- Systèmes hybrides à commutation contrôlée où la variable d'état  $x$  change instantanément en réponse à une entrée externe (commande). Le système constitué d'un réservoir avec une vanne d'alimentation et une vanne d'évacuation présente clairement un cas de système à commutation commandée.
- Systèmes hybrides à saut contrôlé où la valeur de  $x$  change de façon discontinue sous l'influence d'une action extérieure. Le système électrotechnique avec des entrées impulsionnelles illustrent un cas de saut commandé.

### I.5 Autres phénomènes considérés comme système hybride :

La littérature mentionne également d'autres phénomènes pouvant être inclus dans la classe de systèmes hybrides, les phénomènes les plus connus dans le domaine des systèmes hybrides sont décrits [25][26] ci-dessous :

- **Événements de temps** : Événements se produisant à des instants prédéterminés, qui peuvent être traités efficacement car le comportement du système est connu à l'avance.
- **Événements d'état** : Si les événements se produisent à cause du système qui franchissent les valeurs du seuil ; l'instant d'occurrence de l'événement n'est pas connu a priori (comme dans le cas de commutation autonome).
- **Modèle de simulation** : Les équations du système changent dans le cas d'une commutation, un ensemble d'équations peut apparaître, disparaître, ou être remplacé par un autre, ainsi que des contraintes algébriques peuvent être actives ou inactives.
- **Réinitialisation** : Les valeurs des états du système peuvent être modifiées de façon discontinue quand le mode change ou à cause d'un événement.
- **Itération des événements** : Quand un événement se produit, les nouvelles valeurs des variables, à cause de cet événement, peuvent engendrer un autre événement.
- **Chattering** : Si le système présente un phénomène de va-et-vient continu entre deux modes, le système se caractérise par un comportement dit de « **chattering** ».

### I.6 Exemples de classes de systèmes dynamiques hybrides :

#### • Exemple 1 : Balle à rebondissement :

Pour illustrer les systèmes impulsionnels (ou à saut), nous présentons le modèle classique d'une balle à rebondissement comme exemple illustratif.

La dynamique d'une balle à rebondissement de masse  $m$  soumise à l'action de la gravité  $g$  peut être modélisée comme système dynamique impulsionnel. Si la balle est lâchée d'une altitude  $z_0$  avec une vitesse initiale nulle. L'altitude  $z(t)$  de la balle suit donc l'équation différentielle issue de la mécanique classique  $m\ddot{z}(t) = -mg$ . Quand  $z(t) = 0$ , la balle touche le sol et rebondit en perdant une fraction de son énergie :

$$\dot{z}(t^+) = -c\dot{z}(t^-), \text{ avec } c \leq 1$$

En posant  $x_1(t) = z(t)$ ,  $x_2(t) = \dot{z}(t)$ , le modèle hybride de la balle à rebondissement est donné par :

$$H_{br} = \{Q, \{\varepsilon_w\}, X, \Gamma, Inv, R, G\}$$

Avec  $Q = \{1\}$  est l'ensemble fini d'état discrets. Aucune entrée discrète n'est contrôlable. La sortie discrète se résume à l'avancement non observable  $\{\varepsilon_w\}$  parce qu'aucune sortie discrète n'est accessible à la mesure.

Les transitions discrètes  $\Gamma = \{e = (q, q)\}$  sont de type invariant ( $Inv(q) = (x_1 \neq 0)$ ), la condition de garde est donnée par  $G = \{x_1 = 0\}$ . Lors du franchissement des transitions aucune sortie discrète mesurable n'est émise et le vecteur d'état est réactualisé selon  $(x_1, x_2) = (x_1, -cx_2)$ . L'automate hybride modélisant le comportement de la balle bondissante est donnée par la figure I.5.

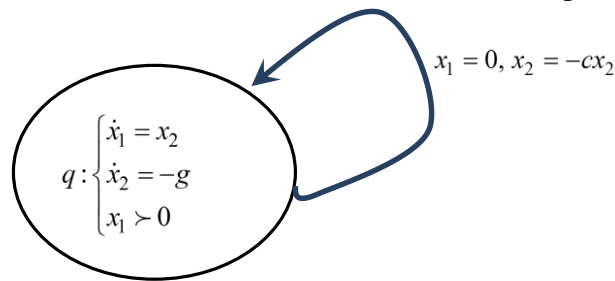


Figure I.5 : Automate hybride de la balle en rebondissement.

### • Exemple 2 : Le thermostat :

Le thermostat utilisé pour maintenir la température constante dans une chambre est un exemple pratique explicite du système linéaire à commutation. Ce système est composé d'un chauffage et d'un capteur de température. Les seuils limites du thermostat sont fixés à des valeurs  $\theta_m$  et, respectivement,  $\theta_M$ , tel que  $\theta_m < \theta_M$ . Le système de chauffage est en marche tant que la température dans la chambre est inférieure au seuil  $\theta_M$ . Le chauffage est arrêté lorsque le capteur détecte le seuil supérieur  $\theta_M$  et il reste à l'arrêt jusqu'au moment où la température chute au-dessous du seuil inférieur  $\theta_m$ .

La température de la chambre et le thermostat peuvent être vus comme un système linéaire à commutations dont l'évolution continue est définie par la variation de la température  $x$  dans la chambre et l'évolution discrète par passage du système de l'état de marche vers l'état d'arrêt et vice versa.

Pour modéliser le système, nous considérons que l'évolution de la température peut être modélisée, par les équations différentielles suivantes :

$$\dot{x} = \begin{cases} f_1(x) = -x + \alpha u & \text{si le chauffage est en marche} \\ f_2(x) = -x & \text{si le chauffage est en arrêt} \end{cases} \quad (I.1)$$

Où  $\alpha \in \mathbb{R}^+$  est une constante réelle positive.

D'une manière graphique le système considéré peut être représenté par un graphe orienté présenté dans la figure I.6., les sommets du graphe correspondent aux dynamiques continues



des états discrets du système. Notamment, la dynamique  $f_1$  est associée au sommet modélisant l'état en marche du système de chauffage ( $q_1$ ), et  $f_2$  au sommet modélisant l'état d'arrêt ( $q_2$ ). Le passage d'un état vers l'autre est modélisé par des arcs étiquetés. Cette représentation graphique est un automate hybride.

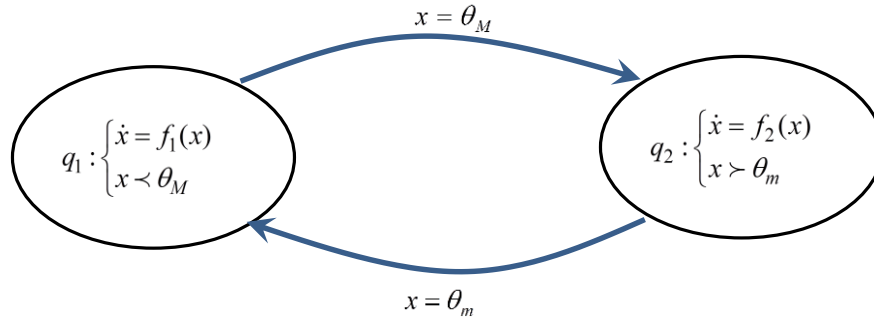


Figure I.6 : Modèle du thermostat.

Les transitions discrètes  $\Gamma = \{e = (q_1, q_2)\}$  sont de type invariant ( $Inv(q_1) = (x < \theta_M)$ ,  $Inv(q_2) = (x > \theta_m)$ ), les conditions de garde sont données par  $G_{q_1} = \{x = \theta_m\}$  et  $G_{q_2} = \{x = \theta_M\}$ . Lors du franchissement des transitions aucune sortie discrète mesurable n'est émise et le vecteur d'état n'est pas réactualisé. L'évolution de la température est donnée par la figure I.7.

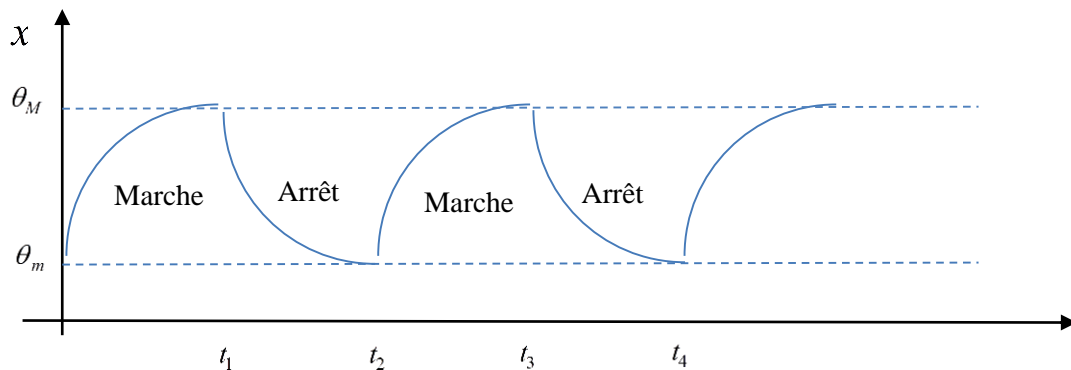


Figure I.7 : Evolution de la température de la chambre.

Le problème d'analyse consiste à vérifier que la température dans la chambre reste toujours dans l'intervalle désiré, notamment :  $\theta_m \leq x \leq \theta_M$ . Les solutions analytiques des équations différentielles peuvent être facilement trouvées. Ainsi, pour une valeur initiale de la température  $x_0 = x(0) = \theta_0$ , les solutions analytiques trouvées sont  $x(t) = \theta_0 e^{-t} + \alpha(1 - e^{-t})$  pour la dynamique correspondant à l'état de marche du système de chauffage, et  $x(t) = \theta_0 e^{-t}$  pour l'état d'arrêt.

Initialement, supposons que le système est dans l'état en marche et la valeur initiale de la température vérifie la relation  $\theta_0 \in [\theta_m, \theta_M]$ . Dans cet état, l'évolution de la température respectera l'expression :

$$x(t) = \theta_0 e^{-t} + \alpha(1 - e^{-t}) \quad (I.2)$$

L'évolution croissante fait que, au bout de  $t_1$  unité de temps, le seuil  $\theta_M$  est atteint. Alors, le système de chauffage passera dans l'état arrête. Suite au changement d'état du système, la dynamique de la température change et la nouvelle évolution est par :

$$x(t) = \theta_0 e^{-(t+t_1)} \quad (I.3)$$

Dans cet état, la température aura une évolution décroissante jusqu'au moment où le seuil inférieur  $\theta_m$  est atteint. A cet instant, le chauffage sera remis en marche et le système reviendra dans l'état initial.

D'après le graphe d'évolution de la température, on constate que ce système représente un système dynamique hybride, il comporte deux types d'évolution, une évolution continue et une évolution discrète (changement d'état discret) interagissant entre eux.

### • Exemple 3 : Système hydraulique :

Le système de la figure I.8.a ci-dessous est composé d'un réservoir d'une section  $S$  associé à une conduite  $C$ . la conduite  $C$  est muni d'une vanne  $V$  avec un comportement Tout ou Rien (TOR : figure I.8.b). Cette dernière permet de commander le passage du liquide. La pompe  $P$  sert pour alimenter le réservoir avec du liquide.

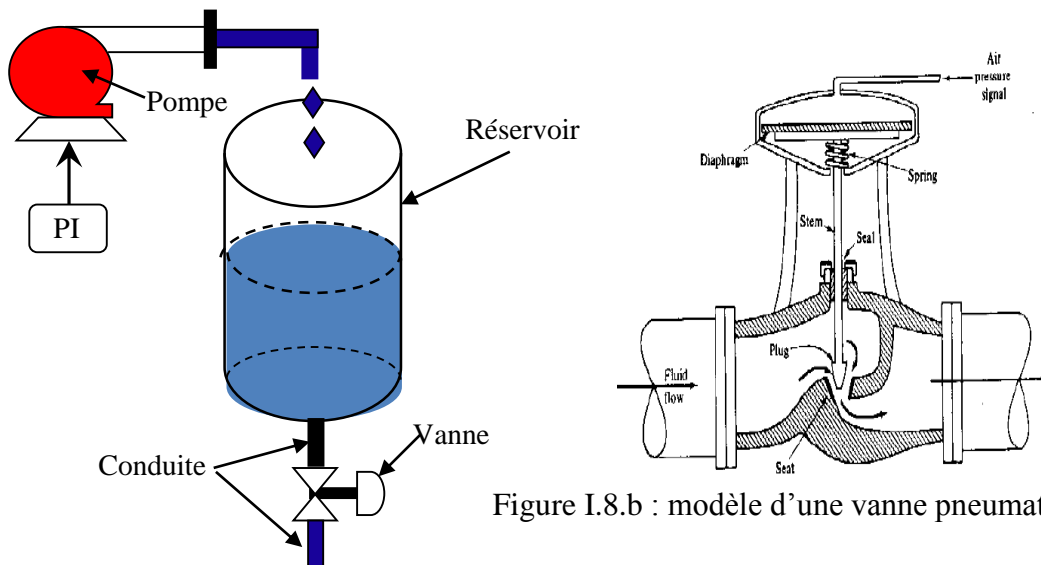


Figure I.8.a : modèle d'un système hydraulique

Le maintien du niveau de liquide à un niveau constant est assuré par l'intermédiaire d'un régulateur PI. Le niveau de liquide  $h(t)$  est une variable dynamique continue. La valeur de la variable  $h(t)$  est fonction des débits entrants (sortie de la pompe) et sortant (évacuation au travers de la vanne  $V$ ) qui ont des évolutions continues.

La fermeture ou l'ouverture est une fonction continue dans le temps donnée par la  $\Phi_e, \Phi_e(x)$  où  $x$  est la position de la tige. Lorsqu'une commande d'ouverture (passage de  $x=0\%$  à  $x=100\%$ ), l'évolution continue de  $x$  donc  $\Phi_e$  est très rapide et la dynamique peut être négligé. Ainsi un

état discret associé à la vanne peut être considéré, et peut prendre dans ce cas deux valeurs ou modalités correspondant respectivement à  $x=0\%$  (vanne fermée) et  $x=100\%$  (vanne ouverte). Ceci suffit en première approximation à caractériser le comportement de la vanne dans le système global. L'introduction de cet état discret met en évidence deux modes de fonctionnement du système et permet de simplifier la modélisation complète du système.

### I.7 Modélisation des systèmes dynamiques hybrides :

Les modèles hybrides sont nécessaires pour répondre à plusieurs problèmes, tels que la définition et le calcul des trajectoires, l'analyse de la stabilité, la synthèse des lois de commande, l'estimation d'état, etc. Plusieurs modèles représentant les systèmes hybrides sont présents dans la littérature, les plus classiquement utilisés sont décrits ci-dessous :

#### I.7.1 Automates hybrides :

L'automate hybride a été présenté par [27][28]. L'automate hybride est constitué d'une machine à état fini. Une dynamique continue est associée à chaque état discret. Un automate hybride consiste en un ensemble de places et d'arcs entre chaque place. Une place représente un mode de fonctionnement et un arc décrit une transition d'un mode de fonctionnement à un autre. A chaque place est associée une inclusion différentielle, et chaque arc est étiqueté par une condition de transition et une relation de saut ou de réinitialisation. A chaque instant, l'état de l'automate hybride est donné par la paire  $(l, x)$  où  $l$  est la place,  $x \in \mathfrak{R}^n$  est l'état continu et  $(l_0, x_0)$  est l'état initial. La trajectoire de l'automate évolue d'une part en fonction des changements instantanés de places, d'autre part en fonction de l'état continu évoluant suivant l'inclusion différentielle dans chaque place.

Un automate hybride est défini par la donnée  $AH = (Q, \Sigma, X, A, inv, F, q_0, x_0)$  tel que :

$Q$  : Ensemble fini représentant les états discrets.

$\Sigma$  : Ensemble d'événements.

$X$  : Ensemble de variables continues à valeurs dans  $\mathfrak{R}^n$ .

$A$  : Ensemble des transitions entre les états discrets.

Dans ce modèle, une transition est définie par un quintuplet  $(q, G, \sigma, R, q')$  avec  $q$  et  $q'$  les sommets source et but respectivement,  $G$  sont les gardes qui déterminent les conditions de commutation d'un état discret à un autre,  $\sigma$  l'événement associé à l'arc et  $R$  un ensemble de fonctions qui permettent de faire évoluer les variables continues pendant les transitions.

$F: Q \times X \rightarrow \rho(X)$  : représente une équation ou inclusion différentielle qui donne l'évolution de la variable continue lorsque l'automate se trouve dans l'état discret correspondant.

$in: Q \rightarrow \varphi(X)$  : Associé à un état une contrainte temporelle appelée « Invariant », qui, tant qu'elle est vraie, permet de rester dans l'état discret.

$q_0$  : Etat initial, et  $x_0$  : valeurs initiales des variables continues.

A chaque instant, comme mentionné précédemment, l'état de l'automate est donné par la paire  $(l, x)$ . L'état de l'automate peut évoluer de deux manières différentes, soit par transition d'un état discret vers un autre état discret en mettant à jour les variables continue, soit par la progression du temps où seules les variables continues sont modifiées.

### I.7.2 Modèle à complémentarité linéaire « LC » :

Un modèle à complémentarité linéaire [8] en temps discret se traduit par les équations suivantes :

$$\begin{cases} x_{k+1} = Ax_k + B_1u_k + B_2w_k \\ y_k = Cx_k + D_1u_k + D_2w_k \\ v_k = E_1x_k + E_2u_k + E_3w_k + g \\ 0 \leq v_k \perp w_k \geq 0 \end{cases} \quad (\text{I.4})$$

Où les vecteurs auxiliaires  $v_k$  et  $w_k \in \mathfrak{R}^s$ , ayant la même dimension, sont orthogonaux. La dernière équation est équivalente à  $v_k^T w_k = 0$ , ce qui implique qu'à chaque instant du temps  $k$  sur la position  $i$  de chaque vecteur  $v_k$  et  $w_k$ , au moins un des éléments  $v_k(i)$  et  $w_k(i)$  est égale à zéro.

Ce modèle est initialement développé pour modéliser les systèmes mécaniques sous contraintes de type inégalité, mais il peut être étendu à la modélisation des systèmes hybrides. Ce modèle peut alors être appliqué à un système mécanique sous contraintes ou avec des relations linéaires par morceaux [29]. La question de l'existence et de l'unicité d'une solution peut être examinée par ce modèle [8].



Figure I.9 : Système composé par deux chariots, exprimé par un modèle hybride de type LC.

Afin de mieux comprendre la signification de vecteurs auxiliaires, on prend l'exemple d'un système composé par deux chariots et deux ressorts sur la figure (I.9). Le premier chariot est attaché au mur à travers un des ressorts et son mouvement est contraint par un obstacle. Les deux vecteurs auxiliaires :  $v_k$  et  $w_k$  représentent respectivement la force de réaction exercée par l'obstacle et la distance  $x_k = w_k$  entre celui-ci et le premier chariot lié au mur. Lorsque le chariot n'est pas en contact avec l'obstacle, la distance  $w_k$  est positive alors que la force  $v_k$  est nulle. En contre partie, quand le chariot est bloqué par l'obstacle, la force réactive est positive et la distance considérée est égale à zéro.

### I.7.3 Modèle à complémentarité linéaire étendue ELC :

Ce modèle est une extension du modèle à complémentarité linéaire [9][10][26]. Il est de type problème de programmation mathématique. Ce modèle peut résoudre certains problèmes fondamentaux de la classe des systèmes à événements discrets en utilisant l'algèbre polynomiale « max-plus », se basant sur la maximisation et l'addition comme opérations de

base. Il peut alors être employé pour modéliser et analyser certaines classes de systèmes hybrides.

On trouve par exemple une utilisation de ce modèle pour traiter le problème de la commande de feux tricolores à une intersection [9].

La classe de ses systèmes hybrides peut être représentée sous la forme mathématique suivante d'un modèle à complémentarité linéaire étendue :

$$x_{k+1} = Ax_k + B_1u_k + B_2d_k \quad (\text{I.5})$$

$$y_k = Cx_k + D_1u_k + D_2d_k \quad (\text{I.6})$$

$$E_1x_k + E_2u_k + E_3d_k \leq g \quad (\text{I.7})$$

$$\sum_{i=1}^p \prod_{j \in \Phi_i} (g - E_1x_k + E_2u_k + E_3d_k)_j = 0 \quad (\text{I.8})$$

Où  $d(k) \in \mathfrak{R}$  est une variable auxiliaire. En raison de l'inégalité (I.7), la condition (I.8) est équivalente avec :

$$\prod_{j \in \Phi_i} (g - E_1x_k + E_2u_k + E_3d_k)_j = 0 \quad (\text{I.9})$$

Pour chaque  $i = \{1, \dots, p\}$ . De ce fait, (I.8)-(I.9) sont considérés comme un système d'inégalités linéaires de type (I.8). Ce système est composé de  $p$  groupes d'inégalités linéaires (un groupe pour chaque ensemble  $\Phi_i$ ) tel que dans chaque groupe il faut qu'au moins l'une d'inégalités soit satisfaite pour égalité :

$\forall i \in \{1, \dots, p\} : \exists j \in \Phi_i$  te que :

$$(g - E_1x_k + E_2u_k + E_3d_k)_j = 0 \quad (\text{I.10})$$

On note qu'on peut toujours exprimer un modèle LC sous la forme étendue d'un modèle ELC par l'introduction de la condition de complémentarité (I.9).

#### I.7.4 Modèle Max-Min-Plus-Scaling MMPS :

La classe de modèles MMPS [10], utilise les opérations : maximisation, minimisation, somme et produit scalaire dans la représentation mathématique des systèmes hybrides. Avant de présenter les équations d'un modèle MMPS, on introduit la définition d'une expression de type « min-max plus scaling ». L'expression MMPS d'une fonction  $f(x_1, x_2, \dots, x_n)$  est donnée par :

$$f \triangleq \chi_i \left| \alpha \left| \max(f_k, f_l) \min(f_k, f_l) \right| f_k + f_l \right| \beta f_k \quad (\text{I.11})$$

Avec :  $i \in \mathbb{N}^*$ ,  $\alpha, \beta \in \mathbb{R}$ ,  $f_k, f_l$  - également des expressions MMPS.

Un modèle MMPS est décrit par :

$$\begin{cases} x_{k+1} = M_x(x_k, u_k, d_k) \\ y_k = M_y(x_k, u_k, d_k) \\ M_c(x_k, u_k, d_k) \leq c \end{cases} \quad (\text{I.12})$$

Où  $M_x, M_y, M_c$  sont des expressions MMPS en fonction de l'état  $x_k$ , de l'entrée commandable  $u_k$  et de la variable auxiliaire  $d_k$ .

### I.7.5 Modèle mixte dynamique et logique "Mixed Logical Dynamical" « MLD » :

Le modèle mixte dynamique et logique « Mixed Logical Dynamical » (MLD) [3] permet de modéliser les systèmes hybrides incluant des états, des entrées, et des sorties mixtes (continus et logiques), ainsi que les systèmes faisant intervenir des relations logiques et des contraintes. Cette structure s'avère très intéressante car elle permet de modéliser plusieurs classes de systèmes hybrides, comme par exemple les systèmes hybrides linéaires, les systèmes linéaires sous contraintes, les systèmes logiques séquentiels (machine d'état fini), quelques classes de systèmes à événements discrets, les systèmes dynamiques non linéaires pour lesquels les non linéarités peuvent être exprimées par l'intermédiaire de combinaisons logiques. De plus, ce type de modèle permet de formuler et de résoudre les problèmes classiques de l'Automatique, tels que la commande de systèmes, l'estimation d'état, l'identification et la détection de pannes [16][30].

Ce type de modèle contient dans sa structure des variables logiques, des dynamiques qui incluent des variables continues et binaires, et des contraintes de fonctionnement. La forme générale d'un modèle logique/dynamique mixte à temps discret est la suivante :

$$x_{k+1} = Ax_k + B_1u_k + B_2\delta_k + B_3z_k \quad (\text{I.13})$$

$$y_k = Cx_k + D_1u_k + D_2\delta_k + D_3z_k \quad (\text{I.14})$$

$$E_2\delta_k + E_3z_k \geq E_4x_k + E_1u_k + E_5 \quad (\text{I.15})$$

Où les états  $x = [x^r \ x^b]^T$ ,  $x^r \in \mathbb{R}^{n_r}$ ,  $x^b \in \{0,1\}^{n_b}$ , les entrées  $u = [u^r \ u^b]^T$ ,  $u^r \in \mathbb{R}^{m_r}$ ,  $u^b \in \{0,1\}^{m_b}$  et les sorties  $y = [y^r \ y^b]^T$ ,  $y^r \in \mathbb{R}^{p_r}$ ,  $y^b \in \{0,1\}^{p_b}$  sont composées des variables continues et binaires.  $\delta \in \{0,1\}^{n_\delta}$  et  $z \in \mathbb{R}^{n_z}$  représentent des variables auxiliaires binaires et continues, introduites en remplaçant des propositions logiques par des inégalité linaires. Les matrices réelles  $A, B_1, B_2, B_3, C, D_1, D_2, D_3, E_1, E_2, E_3, E_4, E_5$  ont des dimensions appropriées. Les équations (I.13), (I.14) et l'inégalité (I.15) contenant toutes les contraintes de fonctionnement sont linéaires. Par conséquent, les non-linéarités du modèle sont formulées à travers les contraintes imposées sur les variables binaires.

### I.7.6 Modèle affine par morceaux PWA :

L'évolution d'un système peut être gouvernée par différents modèles affines selon les valeurs courantes des états et des entrées. Dès lors, l'espace global est divisé en plusieurs régions, chaque région étant définie par des inégalités sur les états et les entrées. La dynamique de chaque région est caractérisée par un modèle affine particulier [31][32].

A cet effet, les diverses études [31][32] ont montré leur intérêt pour la classe de modèles affines par morceaux à temps discret, du fait qu'elle présente plusieurs avantages. Les modèles PWA sont capables de caractériser, d'une manière très précise, de nombreux procédés physiques, tels des systèmes discrets qui présentent des non-linéarités statiques ou des systèmes à

commutation pour lesquels le comportement dynamique est décrit par un nombre fini de modèles linéaires à temps discret et par un ensemble de règles logiques qui sélectionnent le modèle utilisé à chaque instant. En plus, les modèles PWA peuvent approcher des dynamiques non linéaires discrètes au moyen des linéarisations multiples autour des différents points de fonctionnement et également des systèmes hybrides non linéaires à temps continu en utilisant des procédés de discrétisation de la dynamique continue et de remplacement des non linéarités par des approximations affines linéaires. Par ailleurs, la représentation mathématique des modèles PWA facilite l'application de différentes méthodes d'analyse et de synthèse des lois de commande.

Les modèles affines par morceaux sont définies par la partition de l'espace d'état en plusieurs régions polyédrales auxquelles sont associées des fonctions affines de mise-à-jour d l'état et de la sortie. L'expression mathématique en temps discret a la forme suivante :

$$x_{k+1} = A_{j(k)}x_k + B_{j(k)}u_k + f_{j(k)} \quad (\text{I.16})$$

$$y_k = C_{j(k)}x_k + D_{j(k)}u_k + g_{j(k)} \quad (\text{I.17})$$

$$\text{avec } j(k) \text{ tel que : } \begin{bmatrix} x_k \\ u_k \end{bmatrix} \in \mathfrak{R}_{j(k)} \quad (\text{I.18})$$

Où :

- $x = \begin{bmatrix} x^r \\ x^b \end{bmatrix} \in X, X = X_r \times X_b, x^r \in X_r \subseteq \mathbb{R}^{n_r}, x^b \in X_b \subseteq \{0,1\}^{n_b}$
- $u = \begin{bmatrix} u^r \\ u^b \end{bmatrix} \in U, U = U_r \times U_b, u^r \in U_r \subseteq \mathbb{R}^{m_r}, u^b \in U_b \subseteq \{0,1\}^{m_b}$
- $y = \begin{bmatrix} y^r \\ y^b \end{bmatrix} \in Y, Y = Y_r \times Y_b, y^r \in Y_r \subseteq \mathbb{R}^{p_r}, y^b \in Y_b \subseteq \{0,1\}^{p_b}$

représentent les états, les entrées et les sorties réelles et binaires à l'instant  $k \in \mathbb{N}_0$ .  $\mathfrak{R}_{j(k)}$  définissent les différentes régions correspondantes à chaque dynamique dont la partie linéaire est décrite par les matrices réelles  $A_{j(k)} \in \mathbb{R}^{n \times n}, B_{j(k)} \in \mathbb{R}^{n \times m}, C_{j(k)} \in \mathbb{R}^{p \times n}, D_{j(k)} \in \mathbb{R}^{p \times m}$  et la partie affine par les vecteurs réels  $f_{j(k)} \in \mathbb{R}^{n \times 1}$  et  $g_{j(k)} \in \mathbb{R}^{p \times 1}$ . Les dimensions des matrices et vecteurs réels sont exprimées en fonction de :  $n = n_r + n_b, m = m_r + m_b, p = p_r + p_b$

L'ensemble des polyèdres  $\{\mathfrak{R}_{j(k)}\}_{j \in J}$  est défini par des hyperplans dans l'espace des états et des entrées  $X \times U$  :

$$\mathfrak{R}_{j(k)} = \left\{ \begin{bmatrix} x_k \\ u_k \end{bmatrix} : H_{j(k)}^x x_k + H_{j(k)}^u u_k \leq K_{j(k)} \right\} \quad (\text{I.19})$$

tel que :  $H_{j(k)}^x, H_{j(k)}^u$  et  $K_{j(k)}$  décrivent des contraintes linéaires qui déterminent l'espace de chaque région. L'index  $j(k) \in J$  représente le mode du modèle affine par morceaux à l'instant  $k$  du temps et appartient à l'ensemble fini  $J$  contenant tous les modes possibles du modèle. Un mode correspond à une combinaison faisable des états et des entrées binaires.

### I.8 Équivalence entre les modèles hybrides :

Chaque formalisme de modélisation présente ses propres avantages. Les modèles PWA ont été employés afin d'établir des critères de stabilité pour les systèmes hybrides [33][34]. Cependant, les modèles MLD ont servi à l'élaboration de lois de commande [35][36]. L'équivalence les modèles PWA et les autres classes de modèles hybrides (DHA, MLD, LC, ELC, MMPS) est démontrée [37][38]. Cette équivalence permet de transférer les propriétés et des outils théoriques d'une classe de modèles à une autre. A cet effet, le modèle choisi pour la représentation mathématique d'un système hybride pourra être converti vers un modèle équivalent afin de permettre l'analyse et la synthèse des lois de commande (figure I.10).

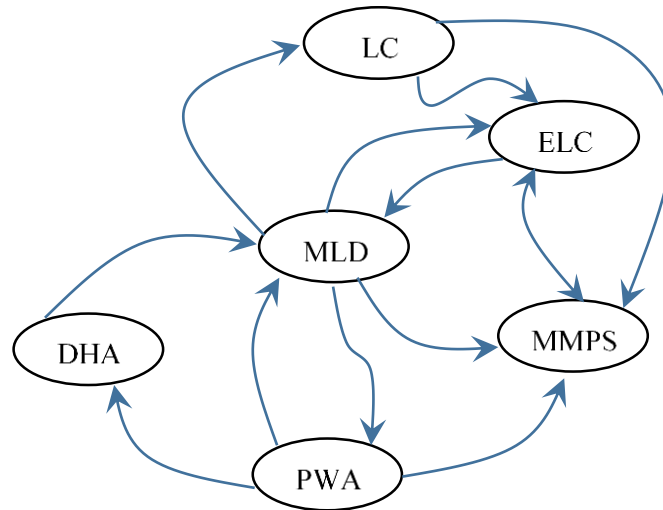


Figure I.10 : Equivalence entre les différents modèles hybrides.

La modélisation des systèmes hybrides est une tâche un peu difficile et pour simplifier l'étape de modélisation, le langage HYSDEL (Hybrid System Description) a été développé [39]. En utilisant ce langage, le modèle d'un système hybride est décrit sous une forme textuelle. De cette manière, HYSDEL permet la modélisation d'une classe des systèmes hybrides discrets définis par l'interaction des systèmes linéaires, des automates et des règles exprimées à travers des propositions logiques. Les descriptions fournies par HYSDEL sont transformées dans des modèles mathématiques discrets MLD. L'utilisation d'une boîte à outils appelée MPT (Multi Parametric Toolbox) [40]. Le modèle MLD obtenu est, soit utilisé directement pour l'analyse et la synthèse de la loi de commande, soit converti vers l'un des modèles équivalents (PWA, LC, ELC, MMPS).

### I.9 Outils de simulation, langages de programmation et logiciels pour les systèmes hybrides :

Plusieurs logiciels et langages de programmation ont été développés pour modéliser et simuler les systèmes hybrides. Dans ce qui suit nous décrivons d'une manière succincte une classification de ces logiciels [41][42].

Les systèmes complexes se composent d'un grand nombre de composant agissant les uns sur les autres avec des comportements non-linéaires et hybrides. La construction précise des



modèles efficaces de simulations pour ces systèmes est une tâche difficile. Des chercheurs ont adopté la programmation orienté-objet pour modéliser de grands systèmes hybrides. Les modèles mathématiques indiquent les différents comportements des composants constituant le système et les modèles formels de calcul définissent des interactions entre les composants, qui fournissent la base pour développer des environnements efficaces pour simuler le comportement des systèmes hybrides.

De nombreux outils de simulation ont vu le jour, ils dépendent généralement du modèle utilisé pour représenter le SDH (bond-graph, automate hybride,...etc.), de leur objectif et du type de simulation utilisée. Parmi ces outils on cite :

- **HYSDEL** (Hybrid system Description Language) est un langage basé sur la modélisation MLD du système hybride. Cet outil peut être employé pour l'analyse et la synthèse de la commande des SDH [39][42].
- **YAHMST** (Yet another Hybrid Simulation Tool) a été développé au laboratoire d'automatique de Grenoble et appliqué au cas d'un processus batch complexe. Il est implanté à l'aide du langage de programmation orienté objet Java et permet la structuration hiérarchique du modèle. Cet outil basé sur l'association d'un modèle discret avec un jeu d'équations continues, il intègre au solveur un détecteur d'événements, facilitant le calcul des instants de commutation [43].
- **HyBrsim** (hybrid Bond graph Simulator) est un outil réalisé sur la base des bonds de graphes hybrides. Cet environnement expérimental de modélisation permet d'établir un cadre formel au SDH considéré [44].
- **HSML** (Hybrid System Modeling Language) : le but de cet outil est de définir formellement le système et de fournir une base pour les langages «front ends » pour des environnements de simulation des systèmes hybrides. Le langage HSML permet une construction hiérarchique et modulaire des modèles, définition du temps continu, du temps discret et des composants logiques. Il permet aussi d'établir un programme prioritaire des composants en temps discret, et des mécanismes pour la manipulation d'état-événement, traitement des conflits, vérification rigoureuse du type et la gamme des variables, et une base sémantique stricte qui permet la vérification et la validation du modèle [45].
- **HSCAP** (hybrid Sequential Causal Assignment Proc.) il permet la mise à jour dynamique de l'information causale. Il est constitué de structures de diagramme de bloc reconfigurable [46].

### **I.9.1 Outils issus des systèmes continus étendus aux systèmes dynamiques hybrides :**

Une première catégorie regroupe spécifiquement des logiciels destinés à modéliser des systèmes continus, augmentés d'éléments à événements discrets. Ces langages ne sont cependant pas conçus pour des systèmes purement à événements discrets. Les éléments à événements discrets permettent alors de modéliser les discontinuités et les actions discrètes de commande. Les outils de cette catégorie sont :

- Dymola
- Omola /Omsim (Object-Oriented Modeling Language)
- HyBrSim (Hybrid Bond gRaph Simulator)
- Simulink/StateFlow
- 20-SIM
- gPROMS (General PROcess Modeling System)
- ABACUSS (Advanced Batch And Continuous Unsteady State Simulator)
- BaSiP (Batch Simulation Package)
- Modelica

### **I.9.2 Outils issus des systèmes à événements discrets étendus aux systèmes hybrides :**

Cette catégorie regroupe les langages spécifiques aux systèmes à événements discrets, auxquels sont ajoutés des éléments fondamentaux des langages modélisant les systèmes continus. Un domaine d'application de cette catégorie est par exemple les procédés par lots où les aspects continus peuvent être la constante ouverte/fermée de débit entre deux réservoirs. Les langages de cette catégorie sont principalement :

- SIMAN (Simulation Manufacturing or Simulation Analysis)
- SiMPLE++(Simulation and Modeling for Process environment based Oriented-Object language)
- Prosim (Process Simulation and Modeling)

### **I.9.3 Outils pour les systèmes dynamiques hybrides (Langages CT/DE) :**

Cette catégorie regroupe les langages de haut niveau incluant des modules issus des langages pour systèmes continus et discrets. Ils peuvent être utilisés pour modéliser des systèmes exclusivement continus, exclusivement discrets ou des systèmes hybrides, permettant de représenter une partie du système physique comme un sous-système continu et une partie comme un sous-système discret. Les langages de cette catégorie sont notamment :

- COSMOS (A Statistical Model Checker for the Hybrid Automata Stochastic)
- Shift (the name is a permutation of HSTIF, Hybrid Systems Tool Interchange Format )
- VHDL-AMS [47]
- Le langage  $\chi$  (ou Chi)

Une autre classification des logiciels et langages existe également, essentiellement basée sur le but recherché (simulation, vérification, spécification). Dans ce qui suit, on donne une description de cette classification :

### **I.9.4 Outils de simulation :**

Les logiciels dans cette catégorie sont :

- ✓ ABACUSS II,
- ✓ DAEPACK,
- ✓ AnyLogic,

- ✓ BaSip,
- ✓ DOORS,
- ✓ Dymola,
- ✓ gPROMAS,
- ✓ HyBrSil,
- ✓ OmSim,
- ✓ SHIFT,
- ✓ Smile, 20-SIM

### **Conclusion**

Dans ce chapitre on a donné une présentation des définitions des systèmes dynamiques hybrides, avec description des types de systèmes : continus, discrets, et hybrides. Nous avons aussi survolé en revue les principales classes de systèmes hybrides considérées dans la littérature, avec les caractéristiques communes qui doivent être prises en considération lors de la phase de modélisation pour donner une présentation uniforme à tous les types de SDH. Par la suite, on a décrit les différents modèles qui existent actuellement pour la modélisation des systèmes dynamiques hybrides. Pour compléter notre étude, un recensement des outils de simulation et d'analyse des SDH a été inclus afin bien cerner le domaine en ce qui concerne les outils disponible en plus le fournisseur et le type de licence pour son obtention. Cette description des outils est justifiée par le fait que le produit utilisé dans notre étude de modélisation et simulation fait partie du tableau récapitulatif de l'annexe A.

## *Chapitre II*

# Modélisation et Simulation du système hybride trois cuves avec l'outil APROS

## II.1 Introduction :

Ce chapitre aborde la modélisation et la simulation sous l'environnement APROS (**Advanced Process Simulation Software**). Cet outil est un logiciel multifonctionnel utilisé par différents procédés tels que les centrales électriques et les réacteurs nucléaires. Il a été développé par **VTT Technical Research Centre of Finland**. Le logiciel **APROS** a été principalement utilisé dans les centrales nucléaires et des centrales de combustion. Il permet de simuler et d'examiner le comportement thermo-hydraulique des procédés au régime transitoire et stationnaire. Il est basé sur un modèle hydrodynamique non homogène et non équilibré pour le système d'écoulement diphasique résolu par des méthodes numérique implicite, semi implicite ou explicite. Le développement et l'enrichissement est en croissance perpétuelle répondant aux exigences des utilisateurs dans différents domaines. La version la plus récente a fait l'objet d'une mutation vers la plateforme Simantics qui inclut des outils de modélisation et simulation tel que Fluent et Modelica.

## II.2 Architecture modulaire d'APROS :

La modélisation est une phase préliminaire dans le processus de simulation. Afin de pouvoir aboutir cette phase, le moteur de simulation support plusieurs modules qui permettent de simplifier cette tâche. L'aspect thermohydraulique d'écoulement gaz/liquide, les circuits d'automatisation et de régulation, les systèmes électriques, et les modèles nucléaires des réacteurs est intégré dans l'environnement de modélisation. A cet effet, une interface graphique pour la conception de modèle de procédé baptisé « Grades » permet de réaliser et de tester nos modèles. Le moteur de simulation contient différents solveurs avec une bibliothèque de modèles prédéfinis en interaction entre eux pour effectuer la simulation dynamique comme est indiquée par la figure II.1.

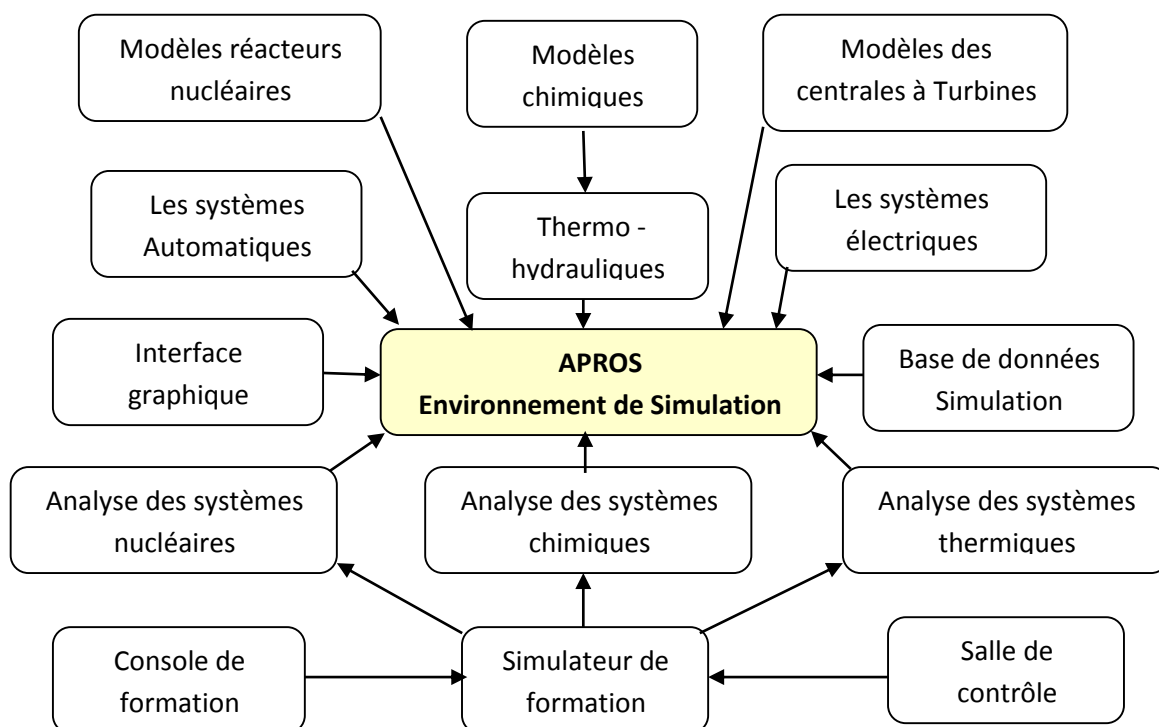
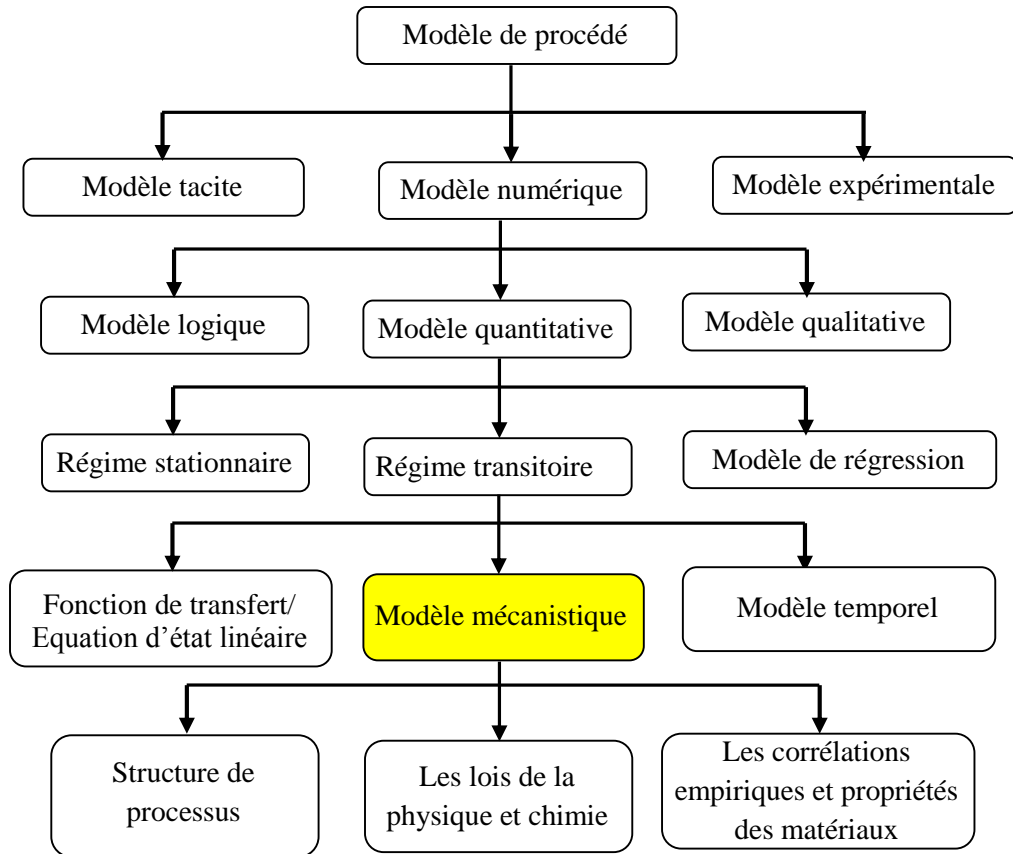


Figure II.1 : Organisation hiérarchique de l'outil de modélisation & Simulation APROS [50].

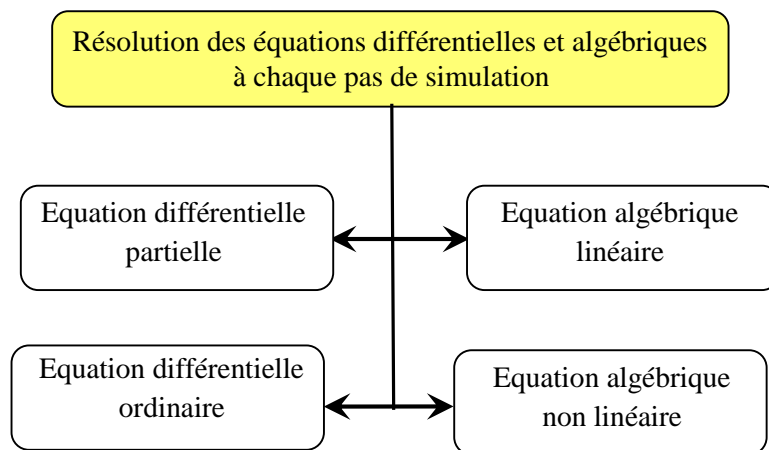
**II.2.1 Architecture dynamique et numérique du code APROS :**

La phase de modélisation sous l’environnement APROS nécessite le recours à différents modèles de processus avec des considérations expérimental. Elle est constituée d’un ensemble de modèle organisé comme l’indique la figure II.2 ci-dessous [50].



**Figure II.2 :** Structure modulaire des calculs transitoires dans APROS.

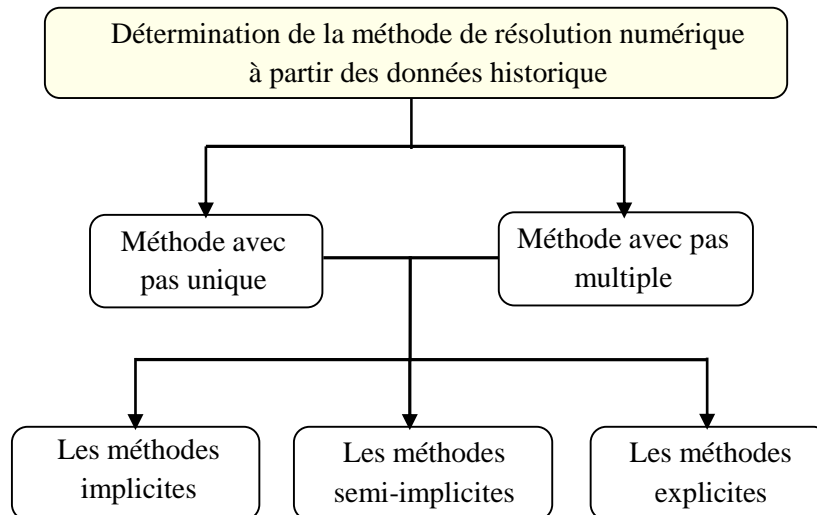
La résolution numérique des modèles élaborés s’effectuent par des méthodes numériques à chaque intervalle de simulation appliquée sur des équations différentielles ou algébriques comme l’indique la figure II.3.



**Figure II.3 :** Architecture différentielle et algébrique du code APROS.

## II.2.2 Modèle hydrodynamique :

Le modèle hydrodynamique du code APROS est un modèle à écoulement unidimensionnel, transitoire à eau-vapeur. Les équations de conservation des deux fluides sont employées comme base pour le modèle hydrodynamique, et qui sont résolues par des méthodes numériques implicites, explicites ou semi-implicites [48][49][50] comme est indiqué sur la figure II.4 ci-dessous.



**Figure II.4 :** Méthode de résolution numérique adoptée par APROS.

Le système décrivant l'écoulement de fluide sous APROS est basé sur un schéma de discrétisation à mailles décalées 'Staggered Grid' [49]. Les équations décrivant l'écoulement de fluide dans un état liquide/vapeur sont données par :

$$\begin{cases} \frac{\partial A\rho}{\partial t} + \frac{\partial A\rho v}{\partial z} = S_1 \\ \frac{\partial A\rho v}{\partial t} + \frac{\partial A\rho v^2}{\partial z} + \frac{A\partial p}{\partial z} = S_2 \\ \frac{\partial A\rho h}{\partial t} + \frac{\partial A\rho v h}{\partial z} = S_3 \end{cases}$$

Où :  $A$  est la section de passage,  $\rho$  est la densité,  $v$  est la vitesse de fluide et  $p$  est la pression. Les termes de droite  $S_1$ ,  $S_2$  et  $S_3$  sont les termes sources de la masse, la quantité de mouvement et l'énergie. Les équations différentielles sont basées sur le concept de volume de contrôle (ou cellule) dans lequel la masse, la quantité de mouvement et l'énergie sont conservées dans le temps et l'espace. Le système d'équation précédent suppose que les propriétés sont moyennées sur la section de passage du fluide. Les propriétés scalaires de l'écoulement (pression, énergie) sont définies aux centres des cellules, les quantités vectorielles (vitesse d'écoulement) sont définies aux frontières des cellules. Les méthodes numériques de résolution des équations différentielles sont utilisées par le calcul de la solution. La méthode implicite est caractérisée par une stabilité sans condition avec un temps plus long. Par contre, la méthode explicite est plus rapide, mais la stabilité est conditionnée. La méthode semi-implicite permet d'avoir les

avantages des deux méthodes, c.à.d. aucune condition sur la stabilité avec un temps de calcul moins long que la méthode implicite.

### **II.2.2.1 Le régime stationnaire :**

Pour effectuer le calcul lié à l'état stationnaire, les algorithmes suivants sont utilisés [48][49]:

- Algorithme pour la cinétique,
- Algorithme pour le système de contrôle,
- Algorithme pour le transitoire hydraulique,
- Algorithme pour le transitoire thermique,

Les paramètres tels que : pressions, densités et débits s'ajouteraient rapidement, mais les effets thermiques évoluent plus lentement. La technique transitoire accéléré, permet de réduire le temps de calcul des transitoires requis pour atteindre l'état stationnaire.

### **II.2.2.2 Le régime transitoire :**

Le calcul transitoire est caractérisé par la variation temporelle d'une ou plusieurs variables liées au problème étudié. Généralement, le régime transitoire doit être précédé par un état stationnaire bien établi dans lequel les conditions initiales sont introduites.

Pour simuler un problème stationnaire ou transitoire, il est nécessaire de définir les valeurs initiales pour chaque composant.

## **II.3 Description de l'environnement de modélisation Grades :**

Grades est un environnement de modélisation industriel graphique fonctionnant sur une variété d'ordinateurs (plateforme software 32/64 bits et configurations (single licence/group licence), il est structuré en modules ordonnés. Des sous programmes constituent le squelette de base pour le traitement thermique, hydraulique et neutronique. La structure de base de l'environnement est composée des éléments suivants [50] :

- Modèles destiné pour la contenir les pages de programmes Nets Grades,
- Les feuilles nets grades pour la synthèse des sous-programmes graphiques à partir des composants de bases (Jonction, valves, conduites, pompes, échangeurs de chaleurs,...etc.),
- Bibliothèque de composants divers et de connexions,
- Utilitaires pour visualiser les données stationnaire et transitoire (graphique ou numérique),
- Modules pour la sauvegarde des données,
- Utilitaire pour l'importation ou l'exportation de modèles.

### **II.3.1 Structure hiérarchique de la modélisation :**

La modélisation consiste à utiliser un ensemble de modèles de composants de processus préfinis qui sont conceptuellement one-to-one avec une forme concrète (pompes, vannes,



réservoirs,...etc.). Le paramétrage des composants s'effectue par l'introduction des propriétés physique. Ensuite, on relie les composants par des connectiques pour établir le lien entre les différents éléments constituant le processus [50].

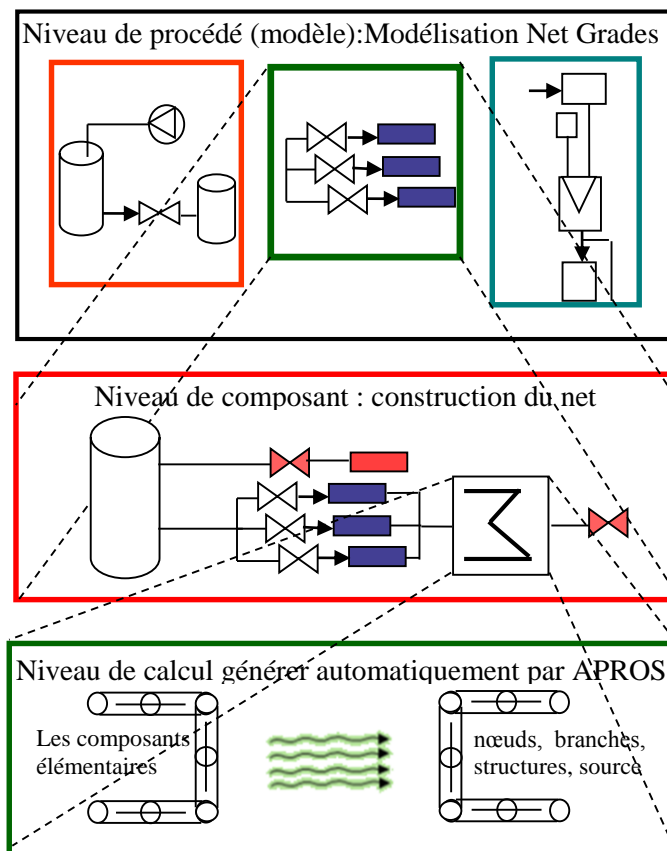


Figure II.5 : Structure hiérarchique du modèle APROS.

La structure de base de données APROS est soumise à un modèle hiérarchique. On agit sur le niveau des composants en utilisant des composants de processus prédéfinis tels que des tuyaux, des vannes, des échangeurs de chaleur, génèrent automatiquement des objets de niveaux de calcul comme est indiqué sur la figure II.5.

### II.3.2 Organisation interne de modèle de procédé :

Le procédé à modéliser est organisé autour d'un ensemble de modèles. Chaque modèle est composé d'une ou de plusieurs sous-modèles contenu dans des pages net Grades. La page net contient les composants élémentaire du procédé (figure ci-dessous figure II.6). Chaque composant dispose d'une interface permettant d'introduire les données d'initialisation, qui par la suite seront utilisés pour la simulation du comportement dynamique. Les composants définis en d'hors de la simulation sont des données déterminant les conditions aux limites.

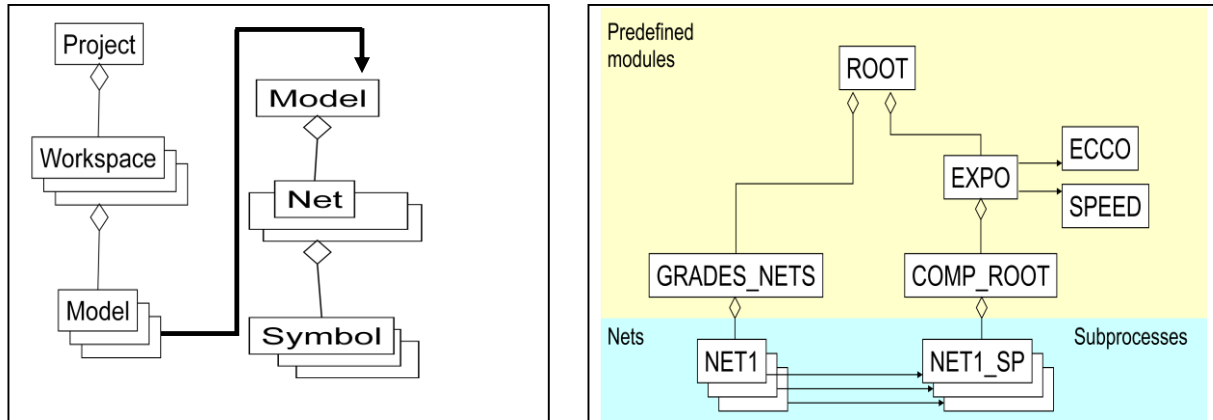


Figure II.6 : Organisation hiérarchique du projet APROS.

### III.3.3 Les mécanisme de communication avec les applications externes :

L'environnement APROS propose plusieurs solutions pour connecter un logiciel externe au moteur de simulation comme l'illustre la figure II.7 ci-dessous.

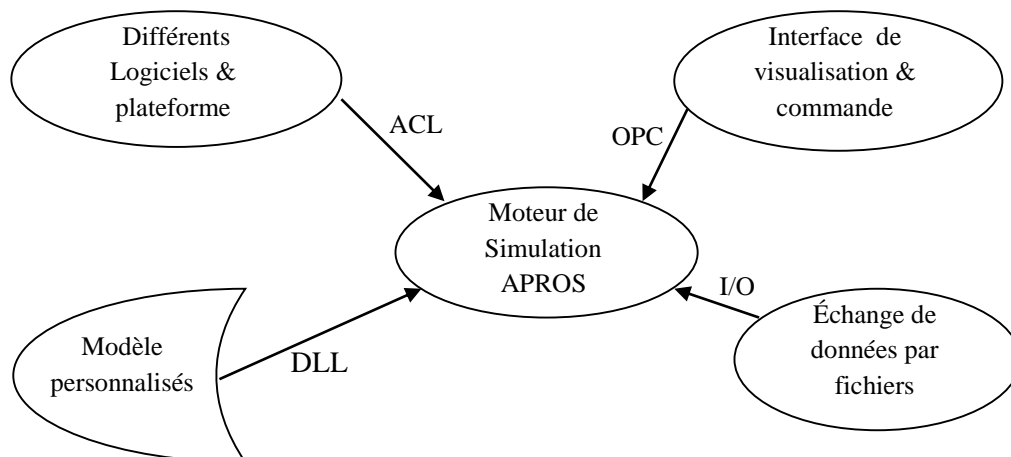


Figure II.7 : Les Interfaces de communications externes supportées par APROS.

Une tierce application peut accéder aux modèles APROS via l'interface OPC (OLE for Process Control). Le mécanisme est principalement destiné à la communication avec les systèmes d'automatisation et d'outils de visualisation, mais il convient pour échanger des données avec d'autres applications qui sont compatibles avec les spécifications d'accès aux données OPC.

La communication entre le serveur OPC et l'applicative client s'effectue au moyen du protocole réseau client/serveur TCP/IP sous l'enveloppe de l'interface OPC automation.

D'autre part, il dispose aussi d'une couche de communication ACL (APROS Communication Library), qui permet de développer des programmes en langage C pour l'échange de données et d'envoyer des commandes vers un processus APROS. Cette couche de liaison est basée sur le protocole TCP/IP, et supporte plusieurs plates-formes matériels et logiciels.

Le développement du programme d'interfaçage entre APROS et MATLAB que nous avons développé dans notre mémoire pour la validation des approches de commande a été essentiellement basé sur le mécanisme OPC.

## II.4 Modélisation des circuits hydrauliques :

### a. circuit hydraulique avec un contrôleur PI de débit :

Dans cette partie, on s'intéresse à la description des étapes élémentaire à la construction d'un modèle sous l'environnement APROS.

L'exemple suivant est composé par deux conduites, une vanne et deux conditions aux limites du problème traité. La connexion des composants est effectuée par l'aide des liens permettant d'établir la relation entre les différents composants comme est illustré sur la figure II.8 ci-dessous.

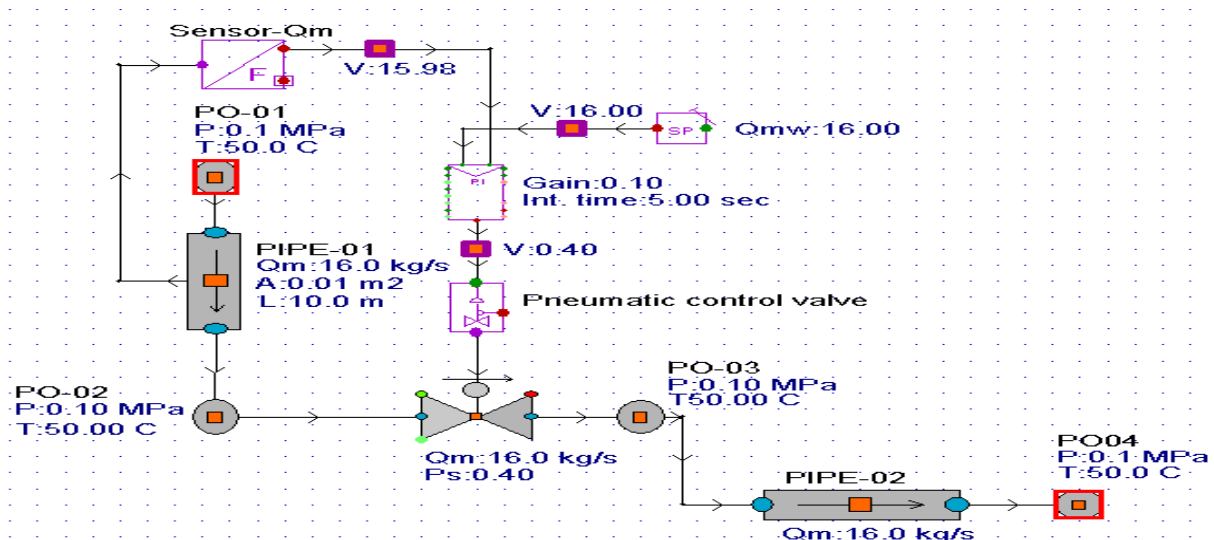


Figure II.8 : Modèle de procédé hydraulique APROS.

Pendant la simulation, APROS résout l'état à chaque pas de temps basé sur les équations de conservation de masse et d'énergie. La détermination du débit massique est basé sur les valeurs initiales, et la variation du débit massique est observée jusqu'à un certain temps nécessaire pour atteindre un état stable ou un état stationnaire.

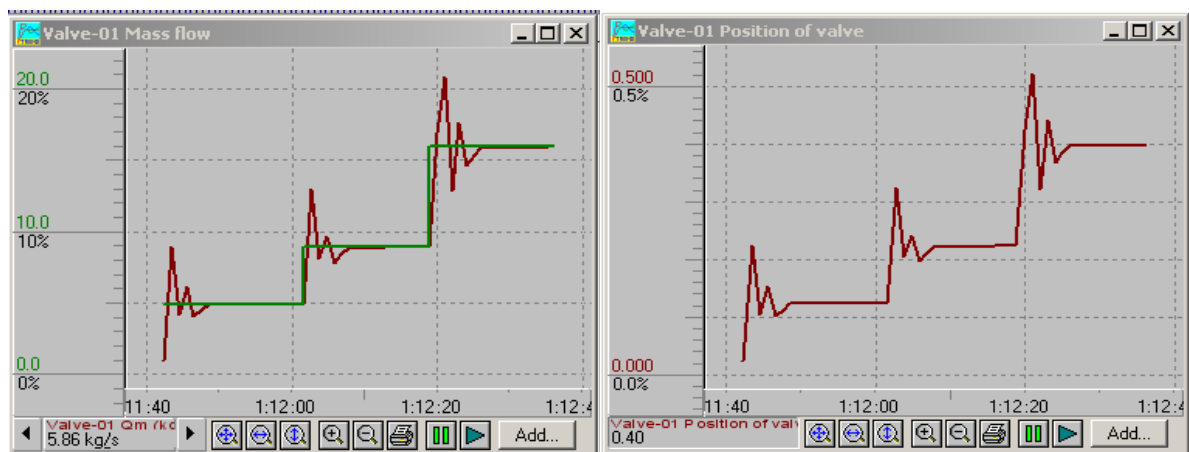


Figure II.9 : Simulation du transitoire avec contrôle de débit du procédé hydraulique (Débit massique & Position de la valve)

Les propriétés des composants en simulation sont calculées sur la base des propriétés des pointes qui sont hors simulation formant les conditions aux limites.

La différence de niveau entre la conduite supérieure et inférieure induit un écoulement de fluide en fonction des propriétés (surface, hauteur, longueur, rugosité).

Le régulateur PI permet le réglage du débit en fonction d'une consigne de référence. La consigne est fixé par un composant dédié du groupe "Automation Analog Basic" et le composant est "SetPoint". La boucle de réglage nécessite un capteur de débit faisant partie du groupe "Automation Measurements" appelé "Flow". Finalement, la commande est appliquée sur la vanne à travers un actionnaire de vanne pneumatique qui fait partie du groupe "Automation Actuators" appelé "Pneumatic control valves". Le tableau suivant liste les composants APROS utilisés dans notre exemple. La simulation du modèle est illustrée par la figure II.9, qui indique la variation de la température en fonction de la position de la vanne.

Description	Composant	Group	
Conduite 1 et 2	PIPE	Pro Pipes	
Jonction 1,2,3 et 4	Point	Pro Point and Node	
Vanne motorisée	Control Valve	Pro Valves	
Capteur de débit	Flow	AUT Measurements	
Signal analogique	Signal Analog	AUT Analog Basic	
Régulateur P.I	PI Controller	AUT Controllers	
Actionneur	A02:Pneumatic Control	AUT Actuators	
Consigne de débit	SetPoint	AUT Analog Basic	

$Qm_w$

**b. circuit avec prédicteur de Smith pour le contrôle de la température :**

L'exemple suivant est composé par un circuit hydraulique avec deux sources d'alimentation de l'eau : Chaud et froid. L'ensemble est rassemblé dans une cuve. Deux vannes permettent d'assurées la fonction de mélange, ou la source froide est fixée à une position de 50% d'ouverture, et la deuxième est contrôlée par l'intermédiaire d'un régulateur de Smith.

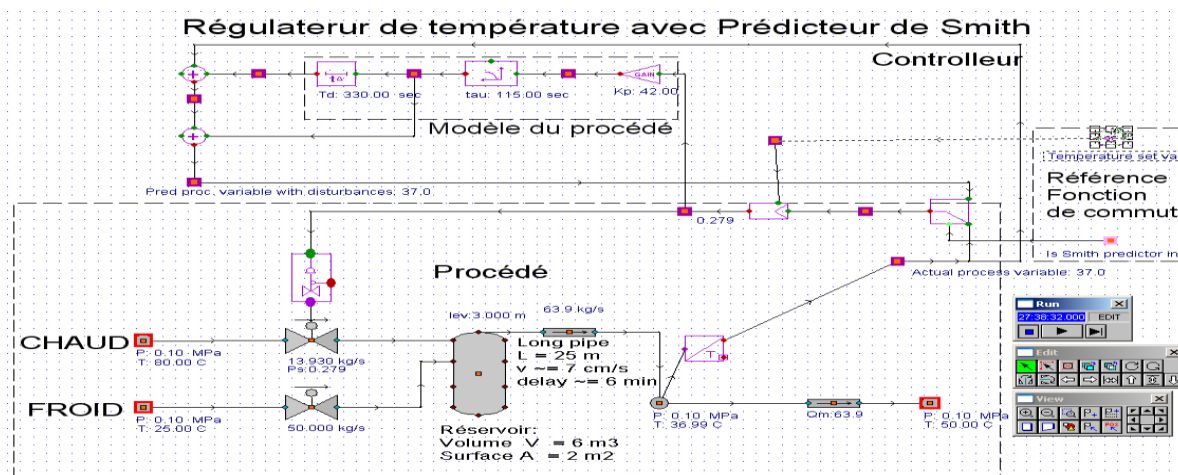


Figure II.10 : Processus thermique avec régulateur de température par le prédicteur de Smith.

La figure II.10 illustre le modèle APROS du contrôleur de Smith.

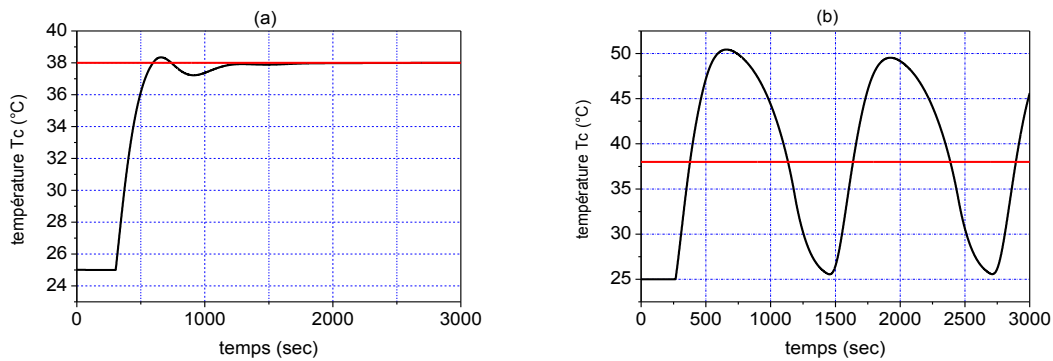


Figure II.11 Simulation du contrôle de température avec (a) et sans (b) prédictiveur de Smith.

### II.5 Description du benchmark COSY « trois cuves » :

Le système hydraulique à trois cuves est régi par un ensemble d'équations mathématiques permettant de décrire son comportement dynamique transitoire et stationnaire :

Le modèle APROS du benchmark représenté par la figure II.12 est, en fait, un modèle destiné pour des problèmes de commande et de détection de pannes de systèmes hybrides [1].

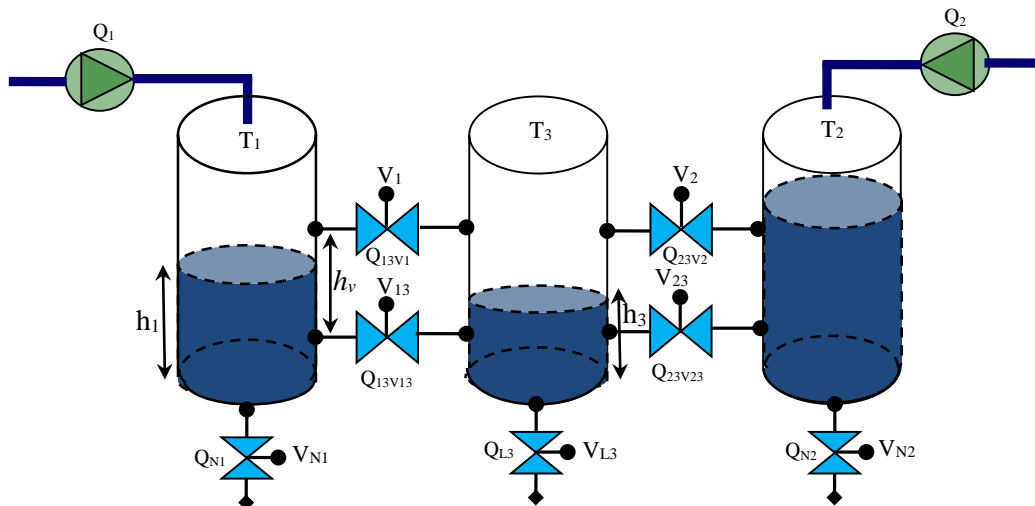


Figure II.12 : Banc d'essai 'Trois cuves' COSY

Le système se compose de trois réservoirs alimentés par deux pompes indépendantes de débit  $Q_1$  et  $Q_2$  dans les réservoirs 1 et 2 respectivement. Ces deux pompes sont manœuvrées de manière continue de 0 jusqu'à un débit maximum  $Q_{max}$ . Les quatre vannes  $V_1$ ,  $V_2$ ,  $V_{13}$  et  $V_{23}$  commandent les débits entre les réservoirs comme est illustré sur la figure V.1. Ces quatre vannes sont de type tout ou rien (ouverte si  $V_i = 1$ , fermée sinon). La vanne manuelle  $V_{N3}$  commande le débit nominal de sortie du réservoir central. On suppose dans toute la suite de notre étude que les vannes manuelles  $V_{N1}$  et  $V_{N2}$  sont toujours fermées et  $V_{N3}$  est ouverte. Les niveaux d'eau à asservir sont notés  $h_1$ ,  $h_2$  et  $h_3$  respectivement.

### II.5.1 Modélisation du système à trois cuves :

La conservation de masse dans les réservoirs fournit les équations différentielles :

$$\begin{cases} \dot{h}_1 = \frac{1}{A}(Q_1 - Q_{13V_1} - Q_{13V_{13}} - Q_{N_1}) \\ \dot{h}_2 = \frac{1}{A}(Q_2 - Q_{23V_2} - Q_{23V_{23}} - Q_{N_2}) \\ \dot{h}_3 = \frac{1}{A}(Q_{13V_1} + Q_{13V_{13}} + Q_{23V_2} + Q_{23V_{23}} - Q_{N_3}) \end{cases} \quad (\text{II.1})$$

Où les variables  $Q_i$  représentent les débits et  $A$  la section de chacun des réservoirs. La loi de Torricelli fournit les expressions des débits dans les vannes :

$$Q_{13V_{13}} = V_{13} a S_{13} \text{sign}(h_1 - h_3) \sqrt{2g(h_1 - h_3)} \quad (\text{II.2})$$

$$Q_{23V_{23}} = V_{23} a S_{23} \text{sign}(h_2 - h_3) \sqrt{2g(h_2 - h_3)} \quad (\text{II.3})$$

$$Q_{13V_1} = V_1 a S_1 \text{sign}(\max(h_1 - h_v) - \max(h_3 - h_v)) \sqrt{2g(\max(h_1 - h_v) - \max(h_3 - h_v))} \quad (\text{II.4})$$

$$Q_{23V_2} = V_2 a S_2 \text{sign}(\max(h_2 - h_v) - \max(h_3 - h_v)) \sqrt{2g(\max(h_2 - h_v) - \max(h_3 - h_v))} \quad (\text{II.5})$$

$$Q_{L1} = V_{L1} a S_{L1} \sqrt{2gh_1} \quad Q_{L2} = V_{L2} a S_{L2} \sqrt{2gh_2} \quad Q_{N3} = V_{N3} a S_{N3} \sqrt{2gh_3} \quad (\text{II.6})$$

Où les  $S_i$  représentent la surface de la vanne  $V_i$  et  $a$  est une constante dépendante du liquide. Les paramètres du modèle sont donnés dans le tableau II.1.

Table II.1- Paramètres du benchmark trois-cuves "Three-tank".

Symbole	Signification	Valeur
$A$	Surface des cuves	<b>0.0154m<sup>2</sup></b>
$a$	Terme correcteur du débit	<b>1</b>
$S_h$	Surface des vannes $V_h$	<b>2.10<sup>-5</sup>m<sup>2</sup></b>
$g$	Constante de gravité	<b>9.81m/s<sup>2</sup></b>
$h_{\max}$	Niveau maximum dans les cuves	<b>0.62m</b>
$h_v$	Hauteur des vanes $V_1$ et $V_2$	<b>0.3m</b>
$Q_{i\max}$	Debit maximum des pompes $Q_i$ $i(i=1,2)$	<b>10<sup>-4</sup>m<sup>3</sup>/s</b>
$T_s$	Période d'échantillonnage	<b>5s</b>

Pour pouvoir modéliser l'ensemble du banc d'essai sous la forme MLD, il convient d'approximer les relations non-linéaires donnant les débits par les équations linéarisées suivantes :

$$Q_{i3V_{i3}} = k_{i3} V_{i3} (h_i - h_3) \quad (\text{II.7})$$

$$Q_{i3V_i} = k_i V_i (\max(h_v - h_i) - \max(h_v - h_3)) \quad (\text{II.8})$$

$$Q_{N3} = k_{N3} V_{N3} h_3 \quad (\text{II.9})$$

Avec :  $i=1,2$ .

$$k_{i3} = aS_{i3} \sqrt{\frac{2g}{h_{\max}}} \tag{II.10}$$

$$k_i = aS_i \sqrt{\frac{2g}{h_{\max} - h_v}} \tag{II.11}$$

$$k_{L1} = aS_{L1} \sqrt{\frac{2g}{h_{\max}}} \quad k_{L2} = aS_{L2} \sqrt{\frac{2g}{h_{\max}}} \quad k_{N3} = aS_{N3} \sqrt{\frac{2g}{h_{\max}}} \tag{II.12}$$

**II.5.2 Modélisation du système à trois cuves sous APROS :**

La modélisation du système COSY sous l’environnement APROS nécessite l’utilisation de certains composants à savoir : des conduites, des vannes TOR piloter par des actionneurs pneumatiques, des réservoirs, des vannes anti-retour, des pompes et des conditions aux limites avec des entrées analogiques et numériques.

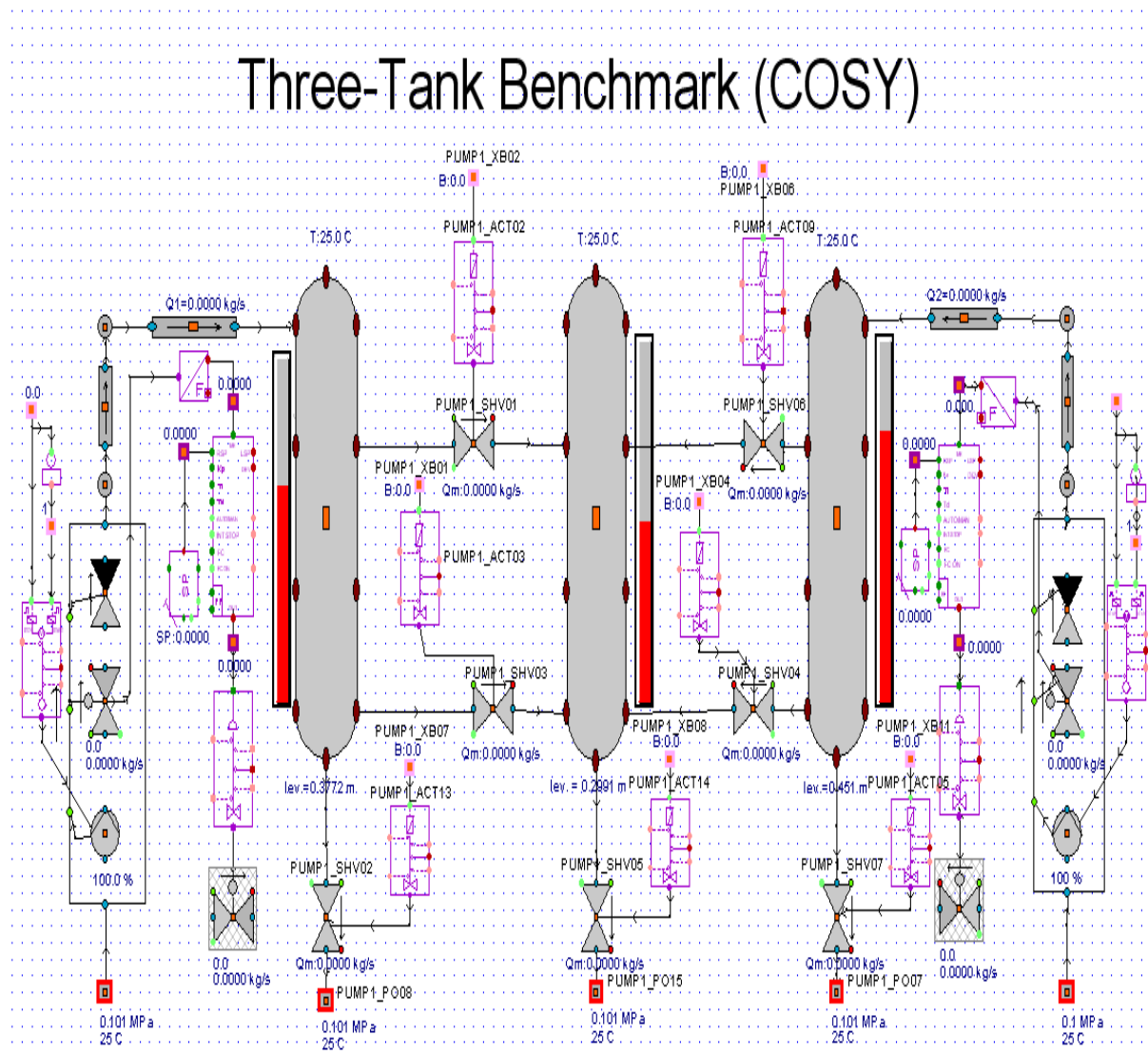


Figure II.13 : Modélisation du système à trois cuves par logiciel APROS.

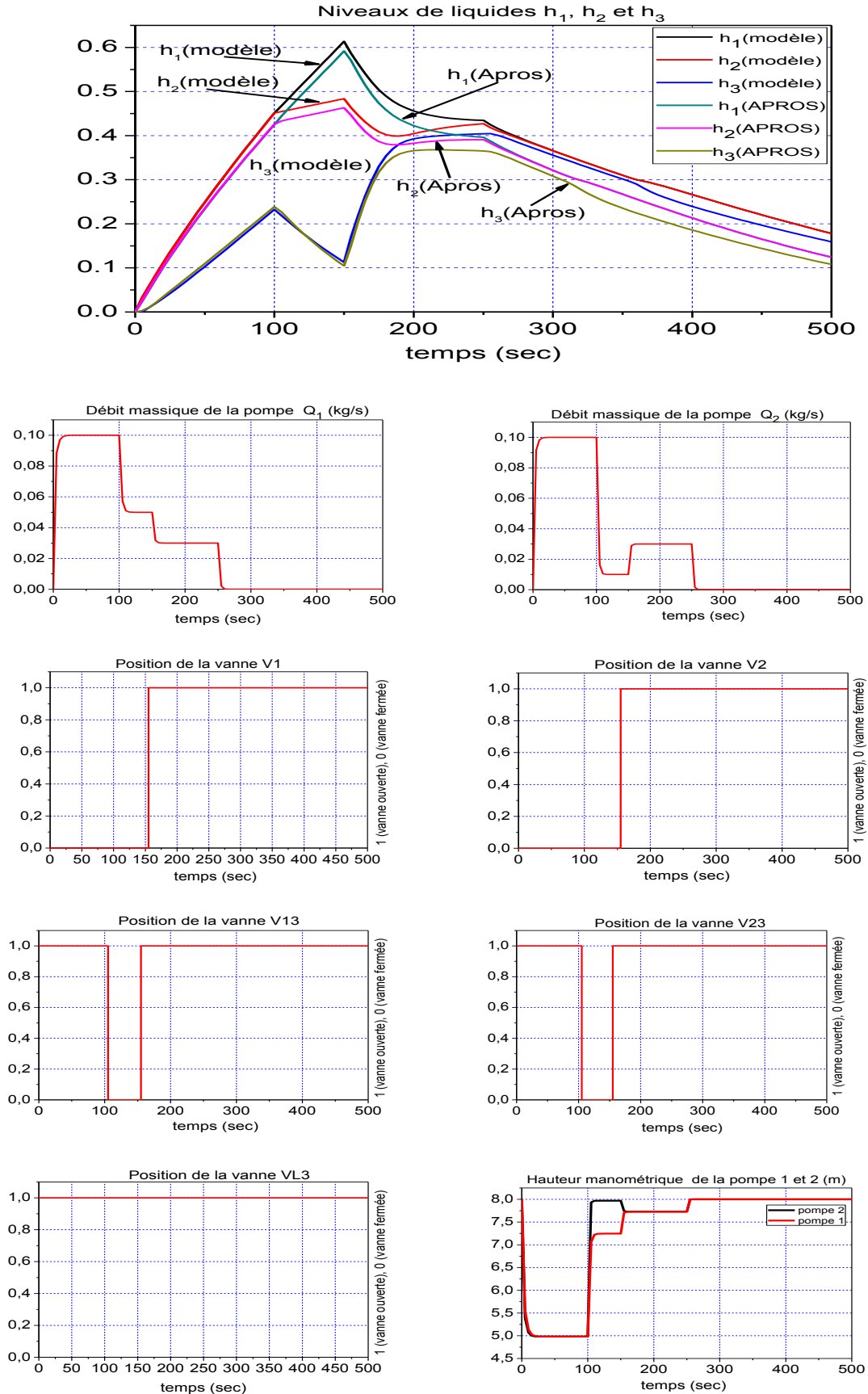


Figure II.14 Simulation en boucle ouverte du système à trois cuves APROS.



La simulation en boucle ouverte effectuée sur notre modèle « trois-cuves » mathématique et APROS, pour des échelons de commande de débit avec variation graduelle entre le débit maximum  $Q_{max}$  et la valeur nulle. Pour les commandes binaires à partir des vannes d'interconnexions entre les réservoirs et de sortie de chacun d'entre eux. Les résultats obtenus montre une certaines différences entre la réponse du modèle mathématique et le modèle APROS qui est due d'une part aux calculs numériques suite à la discrétisation du modèle mathématique et aux paramètres hydrauliques déclarés sous l'environnement d'APROS à savoir la hauteur hydraulique, les pertes de charges engendrés par les vannes, les conduites d'autres part. Cet aspect de modélisation sous l'environnement APROS permet de tenir en compte de tous les considérations physique et hydraulique (la température, la pression, le coefficient de viscosité,...etc.) qui ne sont pas pris en compte dans un modèle mathématique.

### **Conclusion :**

Dans ce chapitre, nous avons présenté l'outil de modélisation et simulation des systèmes dynamiques industriels APROS. Cet outil est à la base de notre application utilisée pour la validation des trois approches de notre contribution scientifique à laquelle sera consacré le chapitre suivant. En premier lieu nous avons présenté l'historique, les fonctionnalités et les applications de cet outil dans le contexte d'un thermo-hydraulicien afin d'appréhender la complexité de modélisation d'un système industriel. Ensuite, nous avons jugé qu'il est important de faire une description de l'environnement de modélisation Grades hiérarchique en survolant d'une manière succincte les étapes de création d'un projet destiné pour contenir un modèle de procédé composé par un ensemble de composants, qui sont les éléments clés dans le processus de modélisation sous l'environnement APROS. Les interfaces d'échanges de données entres les applications externes ont été abordés vu leurs importances dans notre application de commande hybride. Enfin, deux exemples ont été présentés avec simulation suivis par notre modèle trois cuves sujet de l'application du présent mémoire.

## *Chapitre III*

# Commande Prédicative d'un système dynamique hybride sous forme MLD

### III.1 Introduction :

Le problème de commande d'un système hybride consiste à piloter le processus dans un ou plusieurs états souhaités et éviter d'autres états, le tout décrit par des règles de fonctionnalité. De cette façon, nous pouvons considérer les perturbations qui peuvent apparaître dans les modèles des systèmes continus, mais également les événements non contrôlés comme les défauts qui peuvent se produire dans les modèles discrets.

Pour l'analyse du système, on peut être confronté à certaines difficultés qui nécessitent de définir l'interface entre les systèmes continus et discrets. La structure adoptée pour cette définition a une importance prépondérante dans la formalisation globale du système bouclé. D'autre part, les événements modifiant le régime fonctionnel des modèles des systèmes continus ont également un impact sur les valeurs des variables d'état des systèmes. Cet impact se traduit par un saut de fonctionnement des modes continus stables aux modes instables.

Dans ce chapitre nous sommes orientés vers la modélisation et la commande des systèmes décrits par des lois physiques interdépendantes, des règles logiques et des contraintes de fonctionnement, appelés système MLD. Ceux-ci sont décrits par des équations dynamiques linéaires, incluant des inégalités linéaires qui impliquent des variables logiques ou réelles.

Pour ces systèmes, mais généralement pour tous les systèmes hybrides, la forme MLD envisage la modélisation en traduisant les règles logiques en inégalités linéaires. C'est la raison pour laquelle les systèmes MLD regroupant un large ensemble de modèles, parmi lesquels les systèmes hybrides linéaires, les machines d'états finis, quelques classes de système à événements discrets, les systèmes linéaires sous contraintes et les systèmes non linéaires dont les non-linéarités peuvent être exprimées par des fonctions affines par morceaux.

### III.2 Description du formalisme de modélisation MLD :

Dans l'étude présentée dans [3][16][30], on exprime une proposition logique sous forme de contraintes linéaires sur des variables logiques fournit une structure de modélisation puissante, la forme MLD. Ce formalisme permet de décrire un grand nombre de classes importantes de systèmes, comme les systèmes linéaires par morceaux, les systèmes à entrées et états mixtes discrets/continus. Ce cadre inclut de façon prioritaire des contraintes, et incorpore des règles heuristiques dans la description du modèle, en commençant par la description des lois physiques interagissant, des règles logiques et des contraintes de fonctionnement.

Selon les techniques décrites par exemple dans [51][52][53], la logique proportionnelle est transformée en inégalités linéaires impliquant des variables continues et binaires. Ceci fournit des systèmes MLD décrits par des équations dynamiques linéaires et des inégalités linéaires impliquant des variables continues et binaires. Ceux-ci incluent les états physiques/discrets, les entrées continues/logiques et des variables auxiliaires continues/binaires.

### III.2.1 Calcul proportionnel :

Faisant référence à la notation standard [51][52], nous adopterons par la suite les majuscules  $X_i$  pour représenter des relations. La variable  $X_i$  est généralement désigné sous le nom de littéral qui peut avoir une valeur de vérité 'T' « true » ou 'F' « false ».

On peut également associer à un  $X_i$  littéral une variable logique  $\delta_i \in \{0,1\}$ , qui a une valeur 1 si  $X_i=T$  ou 0 autrement.

La programmation entière linéaire « Linear Integer Programming » a été préconisée comme moteur efficace pour effectuer de façon systématique la transformation de relations logiques en inégalités linéaires sur des variables binaires [53]. Un problème de logique proportionnelle, où l'on doit vérifier qu'une relation  $X_1$  est vraie, pour un ensemble de déclarations composées littérales notées  $X_2, \dots, X_n$  peut être en fait résolu au moyen d'une programmation entière linéaire [54]. Ceci peut être obtenu en traduisant convenablement les déclarations composées initiales en inégalités impliquant des variables logiques  $\delta_i$ . En fait, les propositions suivantes et les contraintes linéaires peuvent facilement être vues comme équivalentes [51].

$$\left\{ \begin{array}{l} X_1 \vee X_2 \text{ est équivalente à } \delta_1 + \delta_2 \geq 1 \\ X_1 \wedge X_2 \text{ est équivalente à } \delta_1 = 1, \delta_2 = 1 \\ \sim X_1 \text{ est équivalente à } \delta_1 = 0 \\ X_1 \rightarrow X_2 \text{ est équivalente à } \delta_1 - \delta_2 \leq 1 \\ X_1 \leftrightarrow X_2 \text{ est équivalente à } \delta_1 - \delta_2 = 0 \\ X_1 \oplus X_2 \text{ est équivalente à } \delta_1 + \delta_2 = 1 \end{array} \right. \quad (\text{III.1})$$

'autres méthodes et formulations alternatives pour effectuer la transformation de logique proportionnelle à programmation entière équivalente existent.

Cette technique de calcul d'inférence dépend servira par la suite pour modéliser les parties logiques des processus (commutateurs 'March/Arrêt', réseaux de mécanismes, réseaux combinatoires et séquentiels) et la connaissance heuristique liée à l'opération des systèmes à inégalité linéaires entières.

Se préoccuper des systèmes mixtes dynamiques et logiques impose d'établir un lien entre les deux mondes. A cet effet, il est nécessaire d'établir un lien entre les deux modes ; qui se traduit par des relations entre les événements en entrée et la dynamique physique, on utilisant des inégalités linéaires impliquant des variables continues  $x \in \mathfrak{R}^n$  et des variables logiques (indicateurs)  $\delta \in \{0,1\}$  (« Mixed-Integer Linear Inequalities »).

Considérons,  $X = [f(x) \leq 0]$  où  $f: \mathfrak{R}^n \rightarrow \mathfrak{R}$  est linéaire, et  $x \in \chi$ , où  $\chi$  est un ensemble borné, tel que :

$$\begin{cases} M = \max_{x \in X} f(x) \\ m = \min_{x \in X} f(x) \end{cases} \quad (\text{III.2})$$

Par association d'une variable binaire  $\delta$  à la relation littérale  $X$ , on peut déduire la relation [6] :

$$[f(x) \leq 0] \leftrightarrow [\delta = 1] \text{ est vrai si et seulement si } \begin{cases} f(x) \geq \varepsilon + (m - \varepsilon)\delta \\ f(x) \leq M(1 - \delta) \end{cases} \quad (\text{III.3})$$

Où  $\varepsilon$  est une tolérance (précision de la machine), au-delà de laquelle la contrainte est considérée comme violée.

Par ailleurs, le terme  $\delta f(x)$ , où  $f : \mathfrak{R}^n \mapsto \mathfrak{R}$  et  $\delta \in \{0,1\}$ , peut être remplacé par une variable auxiliaire réelle  $z = \delta f(x)$ , qui satisfait:  $[\delta = 0] \rightarrow [z = 0]$ ,  $[\delta = 1] \rightarrow [z = f(x)]$ . Par conséquent, en définissant  $M, m$  comme dans l'équation (III.2),  $z = \delta f(x)$  est équivalent à :

$$\begin{cases} z \leq M\delta \\ z \geq m\delta \\ z \leq f(x) - m(1 - \delta) \\ z \geq f(x) - M(1 - \delta) \end{cases} \quad (\text{III.4})$$

Tous ces outils seront utilisés pour transformer des faits logiques impliquant des variables continues en inégalités linéaires et exprimer également des relations décrivant l'évolution des systèmes où les lois physiques, les règles logiques et les contraintes de fonctionnement sont interdépendantes.

### III.2.2 Modélisation mathématique par le formalisme MLD :

La modélisation MLD d'un système hybride se base sur la combinaison des trois idées principales :

1. Représenter des relations logiques par des inégalités linéaires sur des variables binaires,
2. Associer des variables logiques et continues via des inégalités linéaires sur des variables binaires et continues,
3. Inclure les variables binaires dans les équations différentielles.

Il est clair que la partie logique peut être facilement transformée en un système d'inégalités. Les propositions logiques sont transformées en inégalités impliquant des variables binaires. Ces propositions peuvent être transcrites en équations traduisant une évolution, comme pour la partie continue, mais avec des contraintes supplémentaires liées aux variables logiques.

#### III.2.2.1 Architecture du formalisme MLD :

Un système hybride est de façon générale composé de deux parties, une partie liée aux dynamiques continues et l'autre aux dynamiques discrètes/numériques. Le schéma ci-dessous décrit le modèle MLD pour un système hybride ainsi que les passages continu/discret associés.

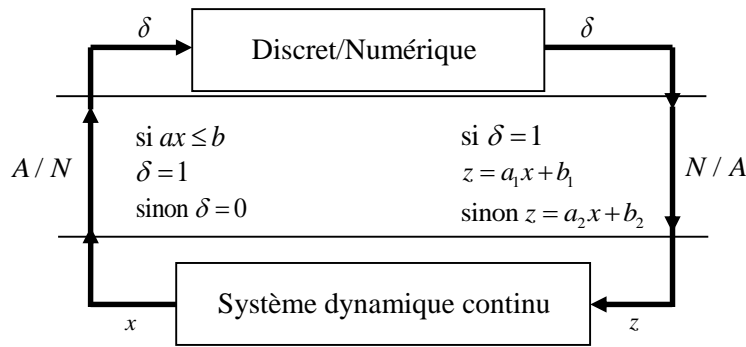


Figure III.1 : Structure d'un modèle MLD.

Des variables auxiliaires permettent de modéliser les relations existantes entre les parties continue et discrète (figure III.1). Ainsi, le passage de la partie continue à la partie discrète nécessite l'ajout de variables logiques. En général, une variable logique auxiliaire est définie pour traduire un phénomène de commutation. Un tel phénomène, on l'a vu préalablement, peut être représenté par une proposition de la forme  $[\delta = 1] \Leftrightarrow [f(x) \leq 0]$ . Pour la partie correspondant au passage discret/continu, des variables auxiliaires continues sont ajoutées, telles que si  $\delta = 1$ , alors  $z = f_1(x)$ , sinon  $z = f_2(x)$ .

### III.2.2.2 Mise en équation du formalisme MLD :

On a vu précédemment que toutes les variables auxiliaires continues et binaires ainsi que les relations logiques peuvent se traduire par des inégalités linéaires sur des variables binaires et continues :

$$[f(x) \leq 0] \Leftrightarrow [\delta = 1] \Leftrightarrow \begin{cases} f(x) \leq M(1 - \delta) \\ f(x) \geq \varepsilon + (m - \varepsilon)\delta \end{cases} \quad (\text{III.5.a})$$

$$z = \delta f(x) \Leftrightarrow \begin{cases} z \leq M\delta \\ z \geq m\delta \\ z \leq f(x) - m(1 - \delta) \\ z \geq f(x) - M(1 - \delta) \end{cases} \quad (\text{III.5.b})$$

Le système sous formalisme **MLD** résultant de l'association continu/discrète sont dès lors décrits par la relation linéaire suivante :

$$\begin{cases} x(k+1) = Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) \\ x(k) = Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k) \\ E_2\delta(k) + E_3z(k) \leq E_1u(k) + E_4x(k) + E_5 \end{cases} \quad (\text{III.6})$$

Cette relation fait intervenir les grandeurs suivantes, mixtes continues/binaires, ou binaires, ou continues :

$$\text{L'état : } x = \begin{bmatrix} x_c \\ x_l \end{bmatrix}, x_c \in \mathfrak{R}^{n_c}, x_l \in \{0,1\}^{n_l}, n = n_n + n_l$$

$$\text{La sortie : } y = \begin{bmatrix} y_c \\ y_l \end{bmatrix}, y_c \in \mathfrak{R}^{p_c}, y_l \in \{0,1\}^{p_l}, p = p_n + p_l$$

$$\text{L'entrée : } u = \begin{bmatrix} u_c \\ u_l \end{bmatrix}, u_c \in \mathfrak{R}^{m_c}, u_l \in \{0,1\}^{m_l}, m = m_n + m_l$$

Les variables binaires auxiliaires :  $\delta \in \{0,1\}^n$

Les variables continues auxiliaires :  $z \in \mathbb{R}^c$

La forme **MLD** représente le système hybride par des équations linéaires, avec inégalités linéaires portant sur les variables continues et binaires. L'ensemble des inégalités de la forme **MLD** rassemble donc d'une part les contraintes du système, d'autre part les inégalités issues des propositions logiques et des variables auxiliaires.

Un problème est dit « bien posé » si la solution en  $\delta$  et  $z$  est unique pour une paire donnée  $(x, u)$ , et par conséquent  $x(k+1)$  est défini de façon unique, ce qui permet de trouver la trajectoire unique des états  $x$ .

### III.3 La commande prédictive :

La prédiction est une notion qui joue un rôle important pour toute activité dans laquelle on cherche à anticiper une trajectoire prédéfinie. De fait, de nombreuses activités humaines telles que la marche, la conduite d'une automobile ou la pratique sportive cherchent à anticiper une trajectoire afin de prévoir les gestes et manœuvres à effectuer. C'est sur ce concept intuitif et naturel que se base la commande prédictive. Les processus industriels, pour une grande partie d'entre eux, doivent également suivre certaines consignes.

Les premiers résultats théoriques et pratiques liés à la commande prédictive ont été obtenus à la fin des années 1970, notamment par les travaux effectués par [55]. Dans les années 1980, plusieurs méthodes basées sur les mêmes concepts prédictifs ont été développées. Parmi, celles-ci, citons la commande prédictive généralisée GPC, développée par Clarke [56], qui s'est avérée la technique la plus largement utilisée par la suite. Une introduction historique sur les diverses méthodes de commande prédictive peut être trouvée dans le livre de Camacho et Bordond [57]. Cette diversité dans la commande prédictive est l'origine de la naissance de la commande prédictive à base de modèle : **Model Predictive Control** ou « **MPC** ».

La commande prédictive à base de modèle MPC a reçu un écho très favorable dans l'industrie parce qu'elle s'avère une technique de commande simple et efficace. La commande prédictive a été implantée dans un grand nombre d'applications industrielles, parmi lesquelles les processus chimiques, qui ont été les premiers à utiliser ce type de commande, les processus de distillation, l'industrie pétrolière et les systèmes électromécaniques. Ces applications industrielles ont toutes un dénominateur commun : la connaissance de la trajectoire à suivre par le système dans le futur, au moins sur un certain horizon. Enfin, ces techniques sont capables de commander une grande variété de processus.

#### III.3.1 Les idées principales de la commande prédictive :

Le but à atteindre par l'utilisation de la commande prédictive est de créer un effet anticipatif exploitant les connaissances explicites sur l'évolution de la trajectoire à suivre dans le futur (connaissances explicites sur l'horizon de quelques points au-delà de l'instant présent). Cette

contrainte permet de tirer parti de toutes les ressources de la méthode ce qui restreint nécessairement le domaine d'application à la commande de système pour lesquels la trajectoire à suivre est parfaitement connue et stockée point par point dans le calculateur.

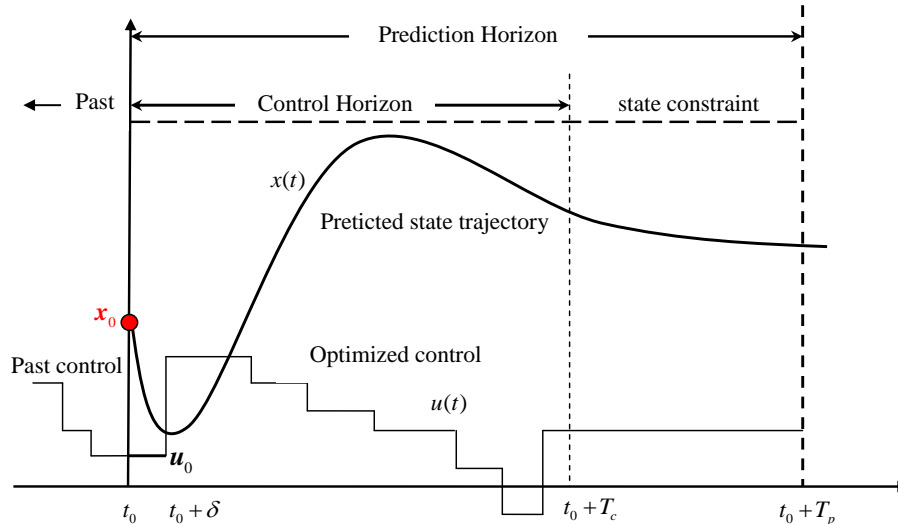


Figure III.2 : Principes de la commande prédictive.

Quatre points sont communs à toutes les méthodes :

1. Définition d'un modèle numérique du système permettant de réaliser la prédiction du comportement futur du système. Ce modèle discret résulte le plus souvent d'une identification préalable hors ligne.
2. Minimisation d'un critère quadratique à horizon fini portant sur les erreurs de prédiction futures, écarts entre la sortie prédite du système et la consigne future ou une trajectoire de référence réalisant un filtre de la consigne (figure III.3).
3. Elaboration d'une séquence de commandes futures, optimale au sens du critère précédent, dont seule la première valeur est appliquée sur le système et le modèle.
4. Répétition des étapes précédentes à la période d'échantillonnage suivante selon le principe de l'horizon glissant.

La commande prédictive, basée sur l'utilisation d'un modèle et le principe de l'horizon glissant (RHC), peut se concevoir comme une stratégie pour laquelle la commande à l'instant  $t$  est obtenue en résolvant en ligne, à chaque instant d'échantillonnage, un problème de commande optimale en boucle ouverte à horizon fini, en utilisant l'état courant du système comme état initial.



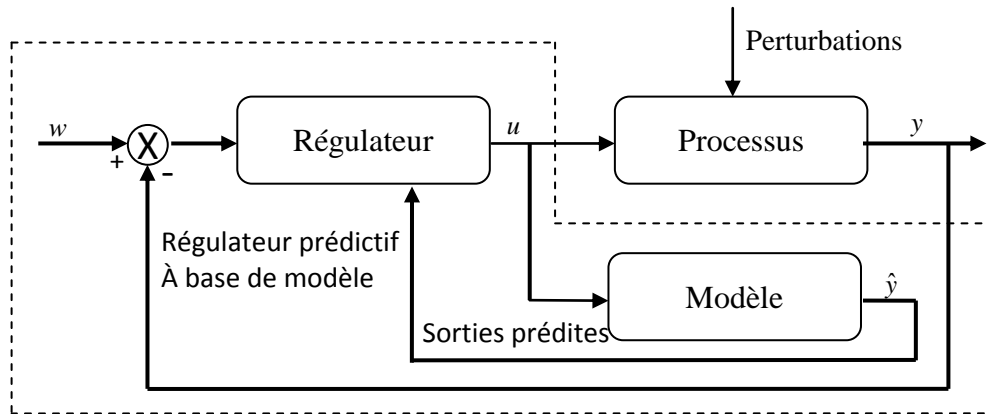


Figure III.3 : Schéma caractérisant la commande prédictive.

La procédure d'optimisation fournit une séquence optimale de commandes futures, dont seule la première est appliquée à l'entrée du processus. On constate l'intérêt de ce type de commande quand la trajectoire à suivre par le système est connue à l'avance.

Le modèle classiquement utilisé est un modèle entrée/sortie de type CARIMA (Controlled AutoRegressive Intergrated Moving Average). La loi de commande GPC est obtenue par minimisation d'un critère quadratique portant sur les erreurs futures avec un terme de pondération sur la commande ou incrément de commande. Des contraintes sur les signaux de sortie peuvent être prises en compte dans le critère.

La liste ci-dessous donne d'une façon non exhaustive les algorithmes classiques de la commande MPC les plus connus :

1. Model Algorithm Control (MAC),
2. Extended Horizon Adaptive Control (EHAC),
3. Extended Prediction Self Adaptive Control (EPSAC),
4. Predictive Functional Control (PFC),
5. Multi-Step Multivariable Adaptive Control (MUSMAR),
6. Multi-Predictor Receding Horizon Control (MURHAC)

Une présentation des algorithmes à base de modèle prédictive control a été proposée par [57].

### III.3.2 Modèle de prédiction pour la commande MPC :

Dans cette partie une description succincte du modèle de prédiction utilisé avec la commande prédictive généralisée qui sera repris par la suite pour l'application aux systèmes hybrides.

Le calcul de la sortie prédite dans le futur requiert l'utilisation d'un modèle numérique du système, que celui utilisé en GPC. Le modèle classiquement utilisé est un modèle entrée/sortie de type CARIMA selon la forme :

$$A(q^{-1})y(k) = B(q^{-1})u(k-1) + \frac{C(q^{-1})}{\Delta(q^{-1})} \xi(k) \quad (\text{III.8})$$

$y(k)$ ,  $u(k-1)$  et  $\xi(k)$  sont respectivement l'entrée, la sortie et le signal de perturbation du système. Le signal  $\xi(k)$  est considéré aléatoire et de moyenne nulle. Le polynôme  $C(q^{-1})$  modélise l'influence du bruit sur le système. L'introduction  $\Delta(q^{-1}) = 1 - q^{-1}$  dans le modèle de bruit a pour but de raisonner sur les incréments de signaux, et par la suite d'apporter une action intégrale dans le correcteur, afin d'annuler toute erreur statique vis-à-vis d'une entrée ou d'une perturbation en échelon.

Le but général est de faire tendre l'erreur de sortie future vers zéro, avec un effort de commande minimum. Ainsi, la loi de commande GPC est obtenue par minimisation d'un critère quadratique portant sur les erreurs futures avec un terme de pondération sur les incréments de commande.

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \beta(j) (\hat{y}(k+j|k) - w(k+j))^2 + \sum_{j=1}^{N_u} \lambda(j) (u(k+j-1))^2 \quad (\text{III.9})$$

Où :  $N_1$ ,  $N_2$  sont les horizons de prédiction inférieur et supérieur sur la sortie,  $N_u$  est l'horizon de prédiction sur la commande,  $\beta(j)$ ,  $\lambda(j)$  sont les pondérations sur la commande. L'horizon de prédiction sur la commande permet de diminuer le nombre de commandes futures calculées puisque l'on suppose la relation suivante :

$$\Delta u(k+j) = 0 \text{ pour } j \geq N_u \quad (\text{III.10})$$

$w(k+j)$  représente la consigne, ou une trajectoire de référence imposant le ralliement de la sortie vers la consigne.

Les contraintes sur la commande, la sortie peuvent être ajoutées au sein de la fonction de cout :

$$\begin{cases} u_{\min} \leq u(k) \leq u_{\max} \\ y_{\min} \leq y(k) \leq y_{\max} \end{cases} \quad (\text{III.11})$$

Dans ce cas, la minimisation devient complexe, ce qui nécessite le recours aux méthodes de programmation quadratique (QP) (ce point sera abordé dans la commande des systèmes hybrides).

### III.4 Commande prédictive d'un système hybride sous forme MLD :

#### III.4.1 Structure du critère d'optimisation :

Pour un système sous forme MLD, la stratégie de commande prédictive MPC développée par [3] peut se résumer de la façon suivante :

Soit  $k$  l'instant courant,  $x(k)$  l'état courant,  $(x_e, u_e)$  un point d'équilibre ou une valeur de référence à atteindre,  $k+N$  l'instant final de prédiction, on cherche à élaborer la séquence de commande future  $u_k^{k+N+1} = \{u(k), \dots, u(k+N-1)\}$  permettant de passer de l'état  $x(k)$  à  $x_e$  en minimisant la fonction de coût suivante :

$$\begin{aligned} \min_{\{u_k^{k+N-1}\}} J(u_k^{k+N-1}, x(k)) &= \sum_{i=0}^{N-1} \|u(k+i) - u_e\|_{Q_1}^2 + \|\delta(k+i/k) - \delta_e\|_{Q_2}^2 + \|z(k+i/k) - z_e\|_{Q_3}^2 \\ &+ \|x(k+i/k) - x_e\|_{Q_4}^2 + \|y(k+i/k) - y_e\|_{Q_5}^2 \end{aligned} \quad (\text{III.12})$$

$$\text{sous les contraintes : } \begin{cases} x(k+N/k) = x_e \\ x(k+i+1/k) = Ax(k+i/k) + B_1u(k+i) + B_2\delta(k+i/k) + B_3z(k+i/k) \\ y(k+i/k) = Cx(k+i/k) + D_1u(k+i) + D_2\delta(k+i/k) + D_3z(k+i/k) \\ E_2\delta(k+i/k) + E_3z(k+i/k) \leq E_1u(k+i) + E_4x(k+i/k) + E_5 \end{cases}$$

$N$  est l'horizon de prédiction sur la sortie,  $\delta_e$  et  $z_e$  les valeurs des variables auxiliaires au point de référence, calculées par résolutions d'un problème MIQP pour l'équation portant sur les inégalités. On pose  $x(k+i+1/k) \triangleq x(k+i, x(k), u_k^{k+i})$ . Par ailleurs, on suppose que :  $Q_i = Q_i^T \succ 0$ , pour  $i = 1, 4$  et  $Q_i = Q_i^T \geq 0$ , pour  $i = 2, 3, 5$ .

Supposons que la solution  $\{u_k^{k+N-1}(j)\}_{j=0, \dots, N-1}$  optimale existe. Selon la philosophie de l'horizon glissant, seule la première valeur  $u(k)$  de cette séquence est appliquée au système.

Les commandes optimales suivantes :  $u(k+1), \dots, u(k+N-1)$  sont alors négligées et le processus d'optimisation complet est répété à l'instant  $k+1$ .

Pour pouvoir examiner plus en détail cette stratégie de commande, il convient au préalable de reformuler le problème d'optimisation quadratique (III.12) initialement proposé par Bemporad et Morari [3] sous une forme similaire au problème GPC [1], permettant de la façon suivante :

$$\begin{aligned} \min_{\{u_k^{k+N-1}\}} J(u_k^{k+N-1}, x(k)) &= \sum_{i=0}^{N-1} \|u(k+i) - u_e\|_{Q_1}^2 + \|\delta(k+i/k) - \delta_e\|_{Q_2}^2 + \|z(k+i/k) - z_e\|_{Q_3}^2 \\ &+ \|x(k+i/k) - x_e\|_{Q_4}^2 + \|y(k+i/k) - y_e\|_{Q_5}^2 \end{aligned} \quad (\text{III.13.a})$$

Sous les relations :

$$\begin{cases} x(k+i+1/k) = Ax(k+i/k) + B_1u(k+i) + B_2\delta(k+i/k) + B_3z(k+i/k) \\ y(k+i/k) = Cx(k+i/k) + D_1u(k+i) + D_2\delta(k+i/k) + D_3z(k+i/k) \\ E_2\delta(k+i/k) + E_3z(k+i/k) \leq E_1u(k+i) + E_4x(k+i/k) + E_5 \end{cases} \quad (\text{III.13.b})$$

En suite les même étapes pour le développement de la commande GPC (CARIMA), de la manière suivante :

$$\begin{aligned} H(z) &= \frac{C}{z} \left( I + \frac{A}{z} + \frac{A^2}{z^2} + \frac{A^3}{z^3} + \dots \right) B_1 + \frac{D_1}{z} \left( I + \frac{A}{z} + \frac{A^2}{z^2} + \frac{A^3}{z^3} + \dots \right) B_2 + \\ &\frac{D_2}{z} \left( I + \frac{A}{z} + \frac{A^2}{z^2} + \frac{A^3}{z^3} + \dots \right) B_2 + \frac{D_3}{z} \left( I + \frac{A}{z} + \frac{A^2}{z^2} + \frac{A^3}{z^3} + \dots \right) B_3 \end{aligned}$$

On remplaçant l'opérateur  $z$  dans l'équation suivante, on obtient :

$$\begin{aligned} y(k) &= D_1u(k) + CB_1u(k-1) + CAB_1u(k-2) + CA^2B_1u(k-3) + \dots + \\ &D_2\delta(k) + CB_2\delta(k-1) + CAB_2\delta(k-2) + CA^2B_2\delta(k-3) + \dots + \\ &D_3z(k) + CB_3z(k-1) + CAB_3z(k-2) + CA^2B_3z(k-3) + \dots + \end{aligned}$$

Les sorties futures peuvent être alors calculées :

$$\begin{aligned} y(k+1) &= D_1u(k+1) + CB_1u(k) + CAB_1u(k-1) + CA^2B_1u(k-2) + \dots + \\ &D_2\delta(k+1) + CB_2\delta(k) + CAB_2\delta(k-1) + CA^2B_2\delta(k-2) + \dots + \\ &D_3z(k+1) + CB_3z(k) + CAB_3z(k-1) + CA^2B_3z(k-2) + \dots + \end{aligned}$$

$$\begin{aligned}
y(k+2) = & D_1 u(k+2) + CB_1 u(k+1) + CAB_1 u(k) + CA^2 B_1 u(k-1) + \dots + \\
& D_2 \delta(k+2) + CB_2 \delta(k+1) + CAB_2 \delta(k) + CA^2 B_2 \delta(k-1) + \dots + \\
& D_3 z(k+2) + CB_3 z(k+1) + CAB_3 z(k) + CA^2 B_3 z(k-1) + \dots +
\end{aligned}$$

Pour un horizon de prédiction  $N$ , on obtient :

$$\begin{aligned}
y(k+N) = & D_1 u(k+N) + CB_1 u(k+N-1) + CAB_1 u(k+N-2) + CA^2 B_1 u(k+N-3) + \dots + \\
& D_2 \delta(k+N) + CB_2 \delta(k+N-1) + CAB_2 \delta(k+N-2) + CA^2 B_2 \delta(k+N-3) + \dots + \\
& D_3 z(k+N) + CB_3 z(k+N-1) + CAB_3 z(k+N-2) + CA^2 B_3 z(k+N-3) + \dots +
\end{aligned}$$

D'une manière plus générale, le terme  $y(k+j)$  ( $j=1,2,\dots,N$ ), l'équation précédente peut être écrite comme :

$$\begin{aligned}
y(k+j) = & \sum_{i=1}^{\infty} CA^{i-1} B_1 u(k+j-i) + D_1 u(j+1) + \sum_{i=1}^{\infty} CA^{i-1} B_2 \delta(k+j-i) + D_2 \delta(j+1) + \\
& \sum_{i=1}^{\infty} CA^{i-1} B_3 z(k+j-i) + D_3 z(j+1) +
\end{aligned}$$

$$\begin{aligned}
y(k+j) = & \sum_{i=1}^j CA^{i-1} B_1 u(k+j-i) + \sum_{i=j+1}^{\infty} CA^{i-1} B_1 u(k+j-i) + D_1 u(j+1) + \\
& \sum_{i=1}^j CA^{i-1} B_2 \delta(k+j-i) + \sum_{i=j+1}^{\infty} CA^{i-1} B_2 \delta(k+j-i) + D_2 \delta(j+1) + \\
& \sum_{i=1}^j CA^{i-1} B_3 z(k+j-i) + \sum_{i=j+1}^{\infty} CA^{i-1} B_3 z(k+j-i) + D_3 z(j+1) +
\end{aligned}$$

$$\begin{aligned}
y(k+j) = & \sum_{i=1}^j CA^{i-1} B_1 u(k+j-i) + \sum_{m=0}^{\infty} CA^{m+j} B_1 u(k-m-1) + D_1 u(j+1) + \\
& \sum_{i=1}^j CA^{i-1} B_2 \delta(k+j-i) + \sum_{m=0}^{\infty} CA^{m+j} B_2 \delta(k-m-1) + D_2 \delta(j+1) + \\
& \sum_{i=1}^j CA^{i-1} B_3 z(k+j-i) + \sum_{m=0}^{\infty} CA^{m+j} B_3 z(k-m-1) + D_3 z(j+1) +
\end{aligned}$$

$$\begin{aligned}
y(k+j) = & \sum_{i=1}^j CA^{i-1} B_1 u(k+j-i) + \sum_{i=1}^j CA^{i-1} B_2 \delta(k+j-i) + \sum_{i=1}^j CA^{i-1} B_3 z(k+j-i) + \\
& D_1 u(j+1) + D_2 \delta(j+1) + D_3 z(j+1) + \\
& \sum_{m=0}^{\infty} CA^{m+j} (B_1 u(k-m-1) + B_2 \delta(k-m-1) + B_3 z(k-m-1))
\end{aligned}$$

$$\begin{aligned}
y(k+j) = & \sum_{i=1}^j CA^{i-1} B_1 u(k+j-i) + \sum_{i=1}^j CA^{i-1} B_2 \delta(k+j-i) + \sum_{i=1}^j CA^{i-1} B_3 z(k+j-i) + \\
& D_1 u(j+1) + D_2 \delta(j+1) + D_3 z(j+1) + \\
& \sum_{m=0}^{\infty} CA^{m+j} (x(k-m) - Ax(k-m-1))
\end{aligned}$$

Après simplification, on obtient :

$$y(k+j) = \sum_{i=1}^j CA^{i-1} (B_1 u(k+j-i) + B_2 \delta(k+j-i) + B_3 z(k+j-i)) + D_1 u(j+1) + D_2 \delta(j+1) + D_3 z(j+1) + CA^j x(k) \quad (\text{III.14})$$

On pose :  $Y = [y(k) \ y(k+1) \ y(k+2) \ \dots \ y(k+N-1)]$  et

$U = [u(k) \ \dots \ u(k+N-1), z(k) \ \dots \ z(k+N-1), z(k) \ \dots \ z(k+N-1)]$ , on obtient la forme compacte suivante :

$$Y = GU + F \quad (\text{III.15})$$

Avec :

$$F = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{N-1} \end{bmatrix} x(k) \quad (\text{III.16})$$

$$F = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{N-1} \end{bmatrix} x(k), \quad G = \begin{bmatrix} D_1 & \dots & 0 & D_2 & \dots & 0 & D_3 & \dots & 0 \\ CB_1 & & & CB_2 & & & CB_3 & & \\ CAB_1 & \ddots & \vdots & CAB_2 & & \vdots & CAB_3 & & \vdots \\ CA^2 B_1 & & & CA^2 B_2 & & & CA^2 B_3 & & \\ \vdots & & & \vdots & & & \vdots & \vdots & \vdots \\ CA^{N-1} B_1 & & D_1 & CA^{N-1} B_2 & & D_2 & CA^{N-1} B_3 & & D_3 \end{bmatrix} \quad (\text{III.17})$$

On utilisant cette formulation dans le critère d'optimisation quadratique :

$$\begin{cases} F(\chi, x(k)) = \min_x \frac{1}{2} \chi^T H \chi + f^T \chi \\ \text{sous les contraintes : } c\chi \begin{cases} = \\ \leq \end{cases} b \end{cases}$$

On aura les matrices suivantes :

$H = P^T QP$ ,  $f = Y_e^T QP$  avec

$$c = \begin{bmatrix} -E_1 & \dots & 0 & E_2 & \dots & 0 & E_3 & \dots & 0 \\ -E_4 B_1 & & & -E_4 B_2 & & & -E_4 B_3 & & \\ -E_4 A B_1 & \ddots & & -E_4 A B_2 & \ddots & & -E_4 A B_3 & \ddots & \\ -E_4 A^2 B_1 & & & -E_4 A^2 B_2 & & & -E_4 A^2 B_3 & & \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ -E_4 A^{N-2} B_1 & \dots & -E_1 & -E_4 A^{N-2} B_2 & \dots & E_2 & -E_4 A^{N-2} B_3 & & E_3 \end{bmatrix}$$

$$b^T = \left[ (E_4 x(k) + E_5)^T \ (E_4 A x(k) + E_5)^T \ \dots \ (E_4 A^{N-1} x(k) + E_5)^T \right]$$

$$P = \begin{bmatrix} B_1 & \cdots & 0 & B_2 & \cdots & 0 & B_3 & \cdots & 0 \\ AB_1 & & & & & & & & \\ \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ A^{N-1}B_1 & \cdots & B_1 & A^{N-1}B_2 & \cdots & B_2 & A^{N-1}B_3 & \cdots & B_3 \\ D_1 & \cdots & 0 & D_2 & \cdots & 0 & D_3 & \cdots & 0 \\ CB_1 & & & CB_2 & & & CB_3 & & \\ CAB_1 & & & CAB_2 & & & CAB_3 & & \\ CA^2B_1 & & & CA^2B_2 & & & CA^2B_3 & & \\ \vdots & & & \vdots & & & \vdots & \vdots & \vdots \\ CA^{N-1}B_1 & \cdots & D_1 & CA^{N-1}B_2 & \cdots & D_2 & CA^{N-1}B_3 & \cdots & D_3 \\ I_{m \times m} & & & & & & & & \\ & I_{m \times m} & & & & & & & \\ & & \cdots & & & & & & \\ & & & I_{m \times m} & & & & & \\ & & & & I_{r_1 \times r_1} & & & & \\ & & & & & \cdots & & & \\ & & & & & & I_{r_1 \times r_1} & & \\ & & & & & & & I_{r_c \times r_c} & \\ & & & & & & & & \cdots \\ & & & & & & & & I_{r_1 \times r_1} \end{bmatrix}$$

**Avec :**  $m = m_c + m_l$

$$Y_e^T = \begin{bmatrix} (Ax(k) - x_e)^T & (Ax(k) - x_e)^T & \cdots & (A^N x(k) - x_e)^T, \\ (Cx(k) - y_e)^T & (CAx(k) - y_e)^T & \cdots & (CA^{N-1}x(k) - y_e)^T, \\ \underbrace{-u_e^T, -u_e^T, \dots, -u_e^T}_N, \underbrace{-\delta_e^T, -\delta_e^T, \dots, -\delta_e^T}_N, \underbrace{-z_e^T, -z_e^T, \dots, -z_e^T}_N \end{bmatrix}$$

$$Q^T = \text{diag} [\text{diag}(Q_4)_N \text{diag}(Q_5)_N \text{diag}(Q_1)_N \text{diag}(Q_2)_N \text{diag}(Q_3)_N]$$

Où  $\text{diag}(Q_i)_N$  est une matrice diagonale d'éléments  $Q_i$  et de dimension  $N$ .

Le nombre de variables binaire intervenant pour l'optimisation est alors égale à  $L = N * (m_l + r_l)$ .

Où  $m_l$  est le nombre de variable binaires de commande (logique/discrète), et  $r_l$  est le nombre de variables binaires auxiliaires.

### III.4.2 Formulation du critère d'optimisation pour la méthode de Bemporad [3][4] :

Pour la formulation du critère d'optimisation MIQP dans le cas où l'horizon de prédiction est différent de celui de commande, on suit la même démarche précédente pour la reformulation du critère suivant sous forme d'un problème MIQP :

$$\min_{u, \delta, z} J \left\{ \begin{array}{l} u_k^{k+N_m-1}, \\ \delta_k^{k+N_p-1}, \\ z_k^{k+N_p-1} \end{array} \right\}, x(k) = \sum_{i=0}^{N_p-1} \|y(k+i/k) - y_e\|_{Q_5}^2 + \|\delta(k+i/k) - \delta_e\|_{Q_2}^2 + \quad (III.18)$$

$$\|z(k+i/k) - z_e\|_{Q_3}^2 + \sum_{i=0}^{N_u-1} \|u(k+i) - u_e\|_{Q_1}^2$$

$$Q_M = 2^* \begin{bmatrix} H_{M1}^T Q_5 H_{M11} & H_{M1}^T Q_5 H_{M2} & H_{M1}^T Q_5 H_{M3} \\ H_{M2}^T Q_5 H_{M1} & H_{M2}^T Q_5 H_{M2} + Q_2 & H_{M2}^T Q_5 H_{M3} \\ H_{M3}^T Q_5 H_{M1} & H_{M3}^T Q_5 H_{M2} & H_{M3}^T Q_5 H_{M3} + Q_3 \end{bmatrix}$$

$$H_{Mi} = \begin{bmatrix} CB_i & 0 & \dots & 0 \\ CAB_i & CB_i & \dots & 0 \\ CA^2 B_i & CAB_i & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{(p-2)} B_i & CA^{(p-3)} B_i & \dots & CB_i & 0 \\ CA^{(p-1)} B_i & CA^{(p-2)} B_i & \dots & CAB_i & CB_i \end{bmatrix} \quad H_{M1} = \begin{bmatrix} CB_1 & 0 & \dots & 0 \\ CAB_1 & CB_1 & \dots & 0 \\ CA^2 B_1 & CAB_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{(m-1)} B_1 & CA^{(m-2)} B_1 & \dots & CAB_1 & CB_1 \\ CA^m B_1 & CA^{(m-1)} B_1 & \dots & CA^2 B_1 & CB_1 + CAB_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ CA^{(p-1)} B_1 & CA^{(p-2)} B_1 & \dots & CA^{(p-m-1)} B_1 & \sum_{i=0}^{p-m} CA^i B_1 \end{bmatrix}$$

whre  $i=2,3$

$$E_{Mi} = \begin{bmatrix} E_i & 0 & \dots & 0 \\ -E_4 B_i & E_i & \dots & 0 \\ -E_4 A B_i & -E_4 B_i & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -E_4 A^{(p-3)} B_i & -E_4 A^{(p-4)} B_i & \dots & E_i & 0 \\ -E_4 A^{(p-2)} B_i & -E_4 A^{(p-3)} B_i & \dots & -E_4 B_i & E_i \end{bmatrix} \quad E_{M4} = \begin{bmatrix} -E_4 \\ -E_4 A \\ -E_4 A^2 \\ \vdots \\ \vdots \\ -E_4 A^{(p-1)} \end{bmatrix} \quad E_{M5} = \begin{bmatrix} -E_5 \\ -E_5 \\ -E_5 \\ \vdots \\ p \text{ times} \\ \vdots \\ -E_5 \end{bmatrix}$$

whre  $i=2,3$

$$E_{M1} = \begin{bmatrix} -E_1 & 0 & \dots & 0 \\ -E_4 B_1 & -E_1 & \dots & 0 \\ -E_4 A^2 B_1 & -E_4 B_1 & \dots & 0 \\ \vdots & \vdots & -E_1 & \vdots \\ -E_4 A^{(m-2)} B_1 & -E_4 A^{(m-3)} B_1 & \dots & -E_4 B_1 & -E_1 \\ -E_4 A^{(m-1)} B_1 & -E_4 A^{(m-2)} B_1 & \dots & -E_4 A B_1 & -E_1 - E_4 B_1 \\ -E_4 A^m B_1 & -E_4 A^{(m-1)} B_1 & \dots & -E_4 A^2 B_1 & -E_1 - E_4 B_1 - E_4 A B_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -E_4 A^{(p-2)} B_1 & -E_4 A^{(p-3)} B_1 & \dots & -E_4 A^{(p-m)} B_1 & -E_1 + \sum_{i=0}^{p-m-1} -E_4 A^i B_1 \end{bmatrix}$$

$$\Phi_M = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ \vdots \\ CA^p \end{bmatrix} \quad f_M = 2 * \begin{bmatrix} x_k^T \Phi_M^T Q_5 H_{M1} - \psi_{sp}^T Q_5 H_{M1} - \mu_{sp}^T \Lambda_{su} + d_k^T S_d^T \Lambda_y H_{M1} \\ x_k^T \Phi_M^T Q_5 H_{M2} - \psi_{sp}^T Q_5 H_{M2} - \delta_{sp}^T Q_2 + d_k^T S_d^T Q_5 H_{M2} \\ x_k^T \Phi_M^T Q_5 H_{M3} - \psi_{sp}^T Q_5 H_{M3} - z_{sp}^T Q_2 + d_k^T S_d^T Q_5 H_{M3} \end{bmatrix}$$

$$S_1 = [E_{M1} \ E_{M2} \ E_{M3}] \quad S_2 = [H_{M1} \ H_{M2} \ H_{M3}] \quad S_3^T = \begin{bmatrix} eye(m(size(B_1, 2))) \\ zeros(size(B_1, 2), p(size(B_2, 2))) \\ zeros(m(size(B_1, 2)), p(size(B_3, 2))) \end{bmatrix}$$

$$A_{in} = \begin{bmatrix} S_1 \\ S_2 \\ -S_2 \\ S_3 \\ -S_3 \end{bmatrix}, \quad \begin{cases} b_1 = E_{M5} - E_{M4} x_k \\ b_{21} = \psi_{max} - \Phi_M x_k - S_d d_k \\ b_{22} = -\psi_{max} + \Phi_M x_k + S_d d_k \Rightarrow b_{in} = \\ b_{31} = \mu_{max} \\ b_{32} = -\mu_{min} \end{cases} \Rightarrow b_{in} = \begin{bmatrix} b_1 \\ b_{21} \\ b_{22} \\ b_{31} \\ b_{32} \end{bmatrix}$$

$$\psi_k = [y_{k+1} \ y_{k+2} \ \dots \ y_{k+p}]^T$$

$$\begin{cases} \mu_k = [u_k \ u_{k+1} \ \dots \ u_{k+m-1}]^T \\ \bar{\delta}_k = [\delta_k \ \delta_{k+1} \ \dots \ \delta_{k+p-1}]^T \\ \bar{z}_k = [z_k \ z_{k+1} \ \dots \ z_{k+p-1}]^T \end{cases} \Rightarrow \chi = [\mu_k \ \bar{\delta}_k \ \bar{z}_k]^T$$

Problème d'optimisation miqp

$$\begin{cases} \min_x \frac{1}{2} \chi^T Q_M \chi + f_M^T \chi \\ \text{subject to } A_m \chi \leq b_m \end{cases}$$

### III.4.3 Programmation de l'algorithme d'optimisation quadratique mixte MIQP :

La mise en place de la commande MPC pour le problème (III.12) exige la solution d'un problème de programmation quadratique mixte (MIQP), c'est-à-dire un problème d'optimisation incluant une fonction de coût quadratique, pour lequel le vecteur d'optimisation se compose de variables mixtes (continues & binaires), et des contraintes linéaires. Le problème d'optimisation (III.12) peut en fait se transcrire sous la forme générique :

$$\begin{cases} F(\chi, x(k)) = \min_x \frac{1}{2} \chi^T H \chi + f^T \chi \\ \text{sous les contraintes : } c\chi \begin{cases} = \\ \leq \end{cases} b \end{cases} \quad (III.19)$$

Où le vecteur d'optimisation est :

$$\chi = [u^T(k), \dots, u^T(k+N-1), \delta^T(k), \dots, \delta^T(k+N-1), z^T(k), \dots, z^T(k+N-1), x^T(k), \dots, x^T(k+N), y(k), \dots, y^T(k+N-1)]^T \quad (III.20)$$



Des techniques de type « Branch and Bound » ont été appliquées avec succès lors des phases d'optimisation [3] du critère (III.14). Plusieurs auteurs s'accordent sur le fait que les méthodes B&B sont les plus efficaces pour résoudre des problèmes MIQP [58]. Le paragraphe suivant propose un résumé de la technique B&B :

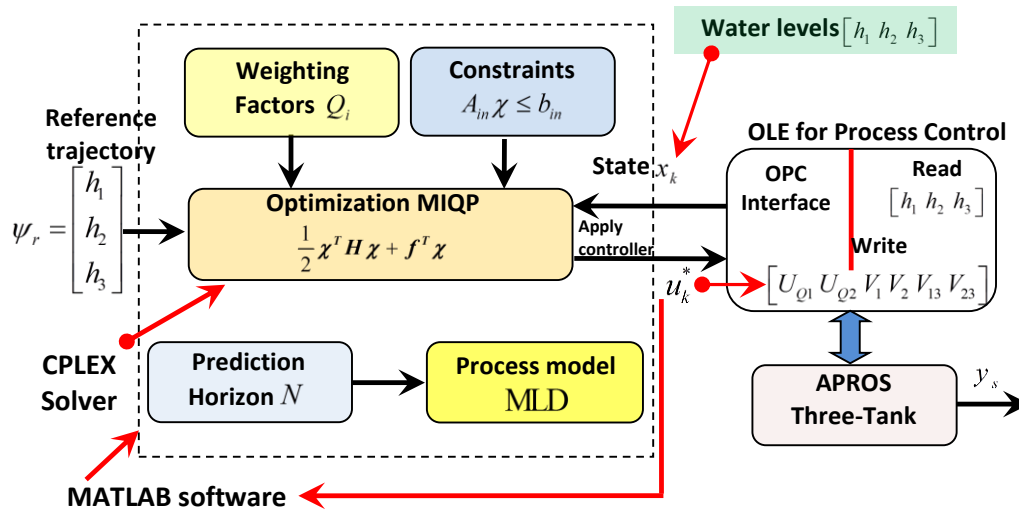


Figure III.4 : Stratégie de commande MPC hybride en ligne appliquée au système à trois-cuves sous l'environnement APROS.

#### III.4.4 Description de l'algorithme d'optimisation MIQP « Branch and Bound » (B&B) :

Toutes les lois de commande qui peuvent être appliquées aux systèmes hybrides sous forme MLD nécessitent le recours à un solveur d'optimisation entier. Deux problèmes importants peuvent être rencontrés « Mixed integer Linear Problems »(MILP) et « Mixed Integer Quadratic Problems »(MIQP).

Dans la littérature, ces types de problèmes sont résolus non seulement pour des variables binaires, mais également pour un domaine de valeurs plus large. Notre cas limite ces problèmes généraux au cas spécifique des variables logique. L'algorithme « **Branch and Bound** » est alors un cadre général permettant de résoudre les problèmes combinatoires et entiers. La partie combinatoire du problème (déterminant la partie optimale entière) est donc résolue par une recherche dans un arbre pendant laquelle des relaxations QP du problème MIQP sont produites et des solutions QP non entières sont éliminées en ajoutant les limites simples (« branch »). En utilisant les limites inférieures et supérieures sur la valeur optimale de l'objectif (« Bound »), il est possible de limiter la recherche dans l'arbre, de ce fait évitant l'énumération complète.

Afin de décrire la stratégie de B&B, il est nécessaire de préciser certaines notations et terminologies. Soit  $P'$  le problème obtenu à partir de  $P$  en relaxant toutes les restrictions entières. Le problème  $P'$  est alors un problème ordinaire QP.

Les deux auteurs [57] ont montré que l'algorithme B&B dans le cadre des problèmes MIQP vise à produire et à résoudre de nouveaux problèmes QP selon un arbre de recherche, où les

nœuds de l'arbre correspondent aux sous-problèmes  $QP$ . L'opération dite « branch » consiste à produire des nœuds enfants à partir des nœuds parents selon des règles de branchement.

Un nœud qui a été entièrement exploré est désigné sous le nom de sondé. Un nœud en attente est un nœud qui a été produit par ramification mais qui n'a pas encore été résolu. L'algorithme B&B explore l'arbre jusqu'à ce que tous les nœuds en attente aient disparu. Il n'est pas toujours nécessaire d'explorer l'arbre complet et le succès du B&B est partiellement dû au fait que des sous-arbres entiers peuvent être exclus de la recherche si leur nœud à la racine produit une solution entière infaisable. La valeur optimale d'un nœud qui donne une solution faisable satisfaisant toutes les contraintes fournit également une limite supérieure sur la solution de 'P'. Cette limite supérieure est alors employée pour sonder les nœuds dont la valeur optimale ou la limite inférieure est supérieure ou égale à la limite supérieure actuelle. Un dernier point indique que l'existence claire de limites inférieure de calcul sur la valeur d'un nœud peut réduire le nombre de problème qui doivent être résolus.

$$\begin{cases} \min x^T Qx + b^T x \\ \text{sous les contraintes : } \begin{cases} Cx + d \leq 0 \\ x = \begin{bmatrix} x_c \\ x_l \end{bmatrix}, x_c \in \mathfrak{R}^{n_c}, x_l \in \{0,1\}^{n_l} \end{cases} \end{cases} \quad (\text{III.21})$$

Il diffère d'un problème  $QP$  standard par la contrainte de type entier, mais également de la stratégie précédente de  $MIQP$  par le domaine des variables binaires. L'idée consistant à résoudre ce  $MIQP$  avec des méthodes de type B&B se fonde également sur la relaxation des contraintes entières, les variables pouvant se trouver dans l'intervalle continu  $[0;1]$ . Un problème ainsi relaxé s'appelle un sous-problème et les valeurs optimales des sous-problèmes, si elles existent, représentent les limites inférieures sur la valeur optimale du  $MIQP$  initial, comme dans la stratégie précédente.

Soit alors  $\xi$  un vecteur de dimension  $n_d$  et le symbole « \* » signifiant que l'entrée correspondant de  $\xi$  est relaxée, c'est dire pouvant se trouver dans l'intervalle continu  $[0;1]$ .

Considérons le problème  $MIQP$  initial sans contrainte entière suivant :

$$\xi_0 = \underbrace{[*,*,\dots,*]}_{n_d} \quad (\text{III.22})$$

Le vecteur  $\xi_0$  sera assigné à la racine de l'arbre. La séparation du problème  $MIQP$  initial en sous-problème  $QP$  relaxés est réalisée en affectant 0 ou 1 à des variables entières. Les nouveaux problèmes  $QP$  résultants sont assignés aux enfants du nœud. Si la composante du  $i^{\text{ième}}$  élément  $\xi^i = 0$  (ou  $\xi^i = 1$ ), alors le  $QP$  correspondant à ce nœud est résolu en plaçant la  $i^{\text{ième}}$  variable binaire à 0 (ou à 1). Comme exemple, considérons un problème  $MIQP$  à trois variables binaires. L'arbre binaire correspondant est donné par la figure III.5 ci-dessous.

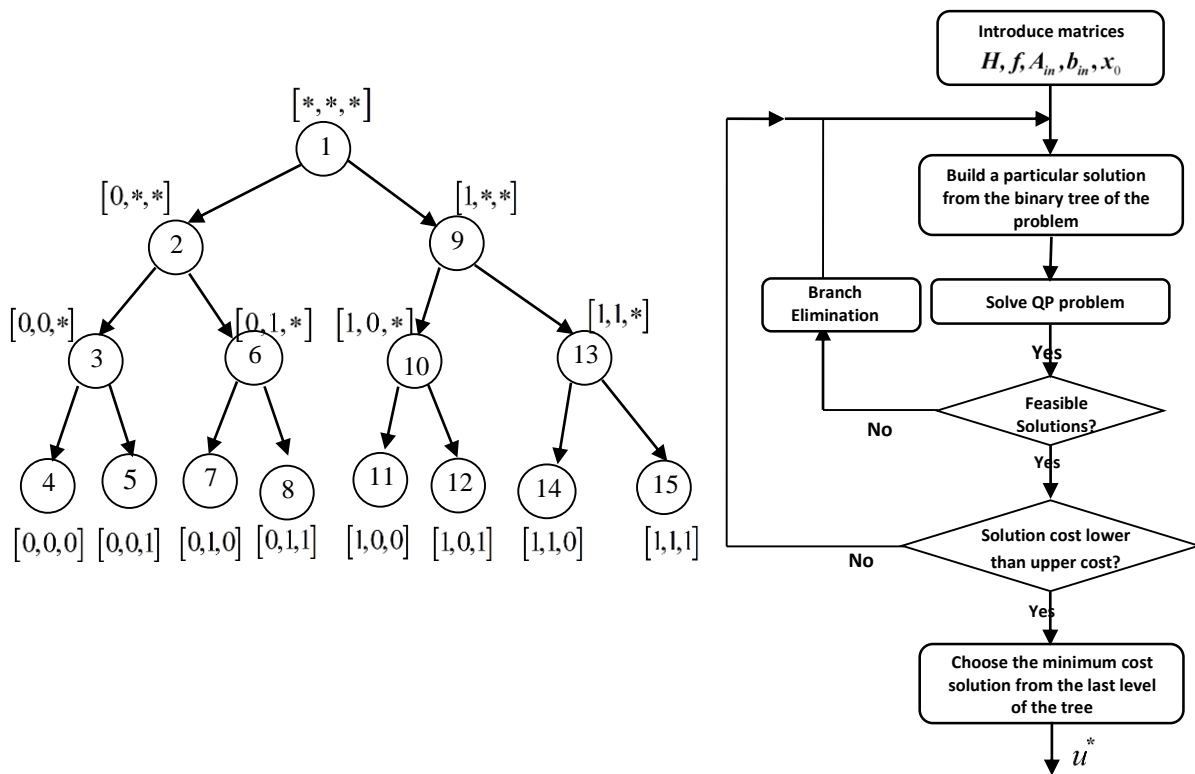


Figure III.5 : Arbre binaire d'un problème *MIQP* à trois variables entières.

Les nombres correspondent à la stratégie de résolution (ici stratégie de type « premier en profondeur »).

L'arbre du problème *MIQP* peut être exploré de plusieurs façons. Le choix de la séparation du problème et de la procédure d'examen des sous-problèmes influence le volume moyen de calcul. Un bon algorithme « Branch&Bound » vise à sonder rapidement les sous-arbres entiers en diminuant le nombre de résolutions de sous-problèmes. Deux choix déterminent l'ordre des sous-problèmes : (1) la règle de branchement, et (2) la stratégie d'exploration de l'arbre.

La règle de branchement choisit la prochaine variable de branchement et décide donc de la façon de séparer les problèmes. Les deux règles de branchement possibles sont :

- ✓ **Première variable libre** : Parmi les variables entières relaxées, il faut choisir celle correspondant au plus petit incrément.
- ✓ **Partie fractionnaire maximale** : En résolvant le problème *QP* relaxé du nœud père, la solution des variables qui devraient être binaires a habituellement une partie fractionnaire. Il faut choisir la variable qui a la plus grande distance à la valeur entière la plus comme prochaine variable de branchement, soit avec l'indice  $i$ , où  $y = \arg \max_i (\min \{ \delta_i, 1 - \delta_i \})$ .

Une fois la variable de branchement choisie par la règle de branchement, la stratégie d'exploration d'arbre (ou de sélection de nœud) détermine la stratégie selon laquelle les problèmes relaxés doivent être résolus. Les stratégies classiques sont les suivantes :

- ✓ **Stratégie du type « premier en profondeur »** : Les problèmes  $QP$  sont résolus par une règle LIFO (last in first out) comme est indiquée sur la figure (III.5).
- ✓ **Stratégie du type « premier selon la largeur »** : Les problèmes de profondeur  $i$  ne sont résolus que si tous les problèmes de profondeur  $i - 1$  ont été résolus, selon la figure (III.6) ci-dessous.

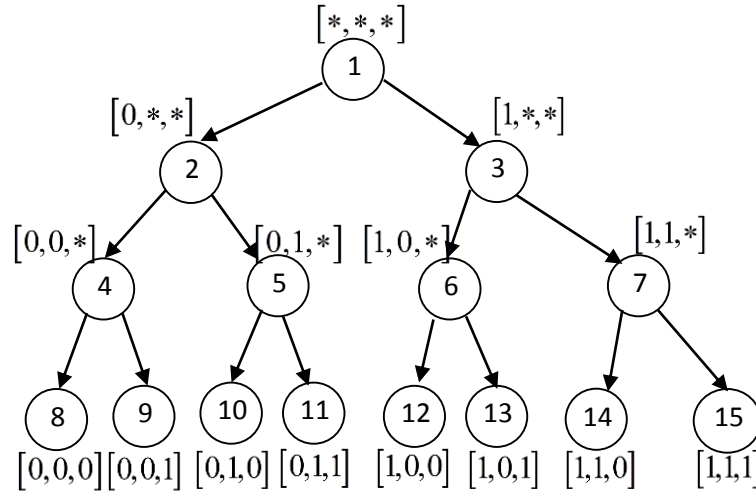


Figure III.6 : Arbre binaire d'un problème  $MIQP$  à trois variables entières.

Les nombres correspondent à la stratégie de résolution (ici stratégie de type « premier selon la largeur »).

Le solveur utilisé pour résoudre le problème  $MIQP$  est développé sous l'environnement MATLAB basé sur une stratégie de type B&B et inclut la stratégie d'exploration de l'arbre, et la règle de sélection des variables de branchement. Le problème résolu par ce programme se présente sous la formulation quadratique  $MIQP$  suivante :

$$\left\{ \begin{array}{l} \min \frac{1}{2} \chi^T H \chi + f^T \chi \\ \text{sous les contraintes : } \end{array} \right. \left\{ \begin{array}{l} A\chi \leq b \\ A_{eq}\chi = b_{eq} \\ v_{lb} \leq \chi \leq v_{ub} \\ \chi \in \mathfrak{R}^{nt_c} \times \{0,1\}^{nt_d} \\ \chi(i_{var\ type}) \in \{0,1\}^{nt_d} \end{array} \right. \quad (III.23)$$

La longueur du vecteur d'optimisation est  $nt = nt_c + nt_d$ . Les variables stockées par l'indice «  $i_{var\ type}$  », sous-ensemble de  $\{1, \dots, nt_c + nt_d\}$ , sont des contraintes binaires. La matrice  $H \in \mathfrak{R}^{nt \times nt}$  est semi-définie positive. Le cas particulier où  $H = 0$  correspond au programme linéaire entier mixte ( $MILP$ ) et peut également être traité par ce programme d'optimisation. La matrice  $A \in \mathfrak{R}^{mt \times nt}$  et le vecteur  $b \in \mathfrak{R}^{mt}$  définissent des contraintes linéaires de type inégalité sur les variables d'optimisation. Des contraintes linéaires de type égalité sont données par  $A_{eq} \in \mathfrak{R}^{mt_{eq} \times nt}$  et  $b_{eq} \in \mathfrak{R}^{mt_{eq}}$ ,

tandis que des limites sur le vecteur d'optimisation peuvent être indiquées par les vecteurs  $v_{lb}, v_{ub} \in \mathbb{R}^{nt}$ .

### III.5 La commande prédictive explicite hors ligne :

Le paragraphe précédent a permis de définir l'ensemble des concepts mettant en œuvre un formalisme global de modélisation de commande à horizon glissant des systèmes hybrides. En effet, il a été montré que la représentation dite « MLD » d'un système hybride rendait possible l'implémentation simple d'algorithmes de commande prédictive. Pour cela, des variables auxiliaires binaires et continues modélisant les interfaces entre les parties discrets et continues ainsi que les opérations logiques ont ajoutées, leur nombre pouvant assez rapidement devenir très important. Il en résulte le principal problème de cette stratégie, lié au temps de calcul de la phase d'optimisation, qui augmente exponentiellement avec le nombre de variables [53]. Ce nombre est proportionnel à l'horizon de prédiction. Ce problème limite fortement l'application pratique de cette stratégie, même si une technique de type « Branch&Bound » est mise en œuvre, et même si l'horizon de prédiction est choisi relativement faible qui par ailleurs minimise l'intérêt d'une architecture prédictive.

Afin de résoudre ce problème, une solution basée sur des techniques de programmation multiparamétrique [59][60][61][62] a été proposée.

Le but de la programmation multiparamétrique est d'obtenir la solution optimale comme une fonction des paramètres du système. Le problème de commande optimale de système modélisé sous forme PWA a ainsi été présenté comme un problème de programmation quadratique multiparamétrique  $mp-LP$ , les entrées et les états du système  $y$  sont traités respectivement comme des variables d'optimisation et des paramètres. On rappelle que les modélisations MLD et PWA sont équivalent [37][38]. La procédure proposée est basée sur l'énumération de toutes les séquences de commutation possibles du système hybride, et ce nombre augmente exponentiellement avec l'horizon de prédiction. L'approche multiparamétrique fournit, par un calcul hors-ligne, un ensemble complet de lois de commande optimales du système en fonction des variables d'état, ainsi que les régions dans l'espace d'état où ses fonctions restent optimales. Il est montré que ces lois sont linéaires et les régions correspondantes sont définies par des inégalités linéaires.

La programmation multiparamétrique linéaire ou quadratique  $mp-MILP$  (ou  $mp-MIQP$ ) vise à optimiser un critère ayant la forme suivante :

$$\begin{cases} V_z(x) = \min_z \frac{1}{2} z' H z + f z \\ \text{sous les contraintes : } G U \leq W + E(S)x \end{cases} \quad (\text{III.24})$$

Où  $z$  représente le vecteur d'optimisation,  $x$  le vecteur des paramètres,  $H = H' \geq 0$  et  $f, G, W, S$  sont des vecteurs et matrices de dimensions appropriées.

La solution obtenue par cette technique est une commande PWA par retour d'état de la forme suivante :

$$u^*(x(k)) = F_i^k x(k) + G_i^k \text{ si } x(k) \in \mathfrak{R}_i^k \quad (\text{III.25})$$

Où  $x \in \mathfrak{R}^n$  est le vecteur d'état,  $u^* \in \mathfrak{R}^m$  est la commande optimale,  $F_i \in \mathfrak{R}^{m \times n}$ ,  $G_i \in \mathfrak{R}^{m \times 1}$  et  $\mathfrak{R}_i^k, i=1, \dots, N_i$  ( $N_i$  nombre de régions polyédrale) définissant la partition de l'espace faisable de  $x(k)$ .

Ces actions optimales de commande sont calculées hors-ligne comme une fonction des variables d'état, et l'espace des variables d'état est subdivisé en régions caractéristiques. Les seuls calculs en ligne consistent, à partir des mesures courants, à identifier la région caractéristique correspondant à ces mesures (à partir de la cartographie précédente élaborée hors-ligne), et à appliquer la loi de commande correspondant à cette région. Le problème d'optimisation est ainsi réduit à un problème simple de fonction d'évaluation.

Ainsi, la loi de commande explicite issue des méthodes de programmation multiparamétrique représente une fonction affine par morceaux, dépendante de la valeur de l'état  $x(k)$  à chaque pas d'échantillonnage. La structure de cette commande par retour d'état est stockée dans trois tables de recherche. La première contient les partitions des régions  $\mathfrak{R}_i^k$  sur lesquelles la commande  $u^*(x(k))$  est définie, alors que la deuxième et la troisième contiennent les gains  $F_i^k, G_i^k$  associés à chaque région du domaine de fonctionnement  $\mathfrak{R}_i^k$ .

L'implémentation de la loi de commande revient à l'identification de la région  $\mathfrak{R}_i^k$  où l'état se trouve à l'instant  $k$  de temps et ensuite au calcul de la commande  $u^*(x(k))$  par l'évaluation de la fonction affine  $f(x(k)) = F_i^k x(k) + G_i^k$ .

Les étapes effectuées à chaque période d'échantillonnage afin de déterminer la valeur de la commande  $u^*(x(k))$  sont les suivantes :

- ✓ utiliser l'état échantillonné à chaque instant pour la recherche de la région qui contient cet état,
  - ✓ s'il n'existe pas une région faisable, alors il faut reconsidérer la synthèse de la commande (voir le réajustement des matrices de pondération, la reformulation des contraintes ou des objectifs de la commande à travers un autre critère d'optimisation).
- En outre, si une région  $\mathfrak{R}_i^k$  active est trouvée, alors l'index  $j$  est utilisé afin de sélectionner les lignes  $j$  des tables qui contiennent les gains  $F_i^k, G_i^k$ ,
- ✓ déterminer la valeur de la commande à l'instant courant  $u^*(x(k)) = F_i^k x(k) + G_i^k$  et l'appliquer au système,
  - ✓ échantillonner un nouvel état et reprendre l'étape 1.

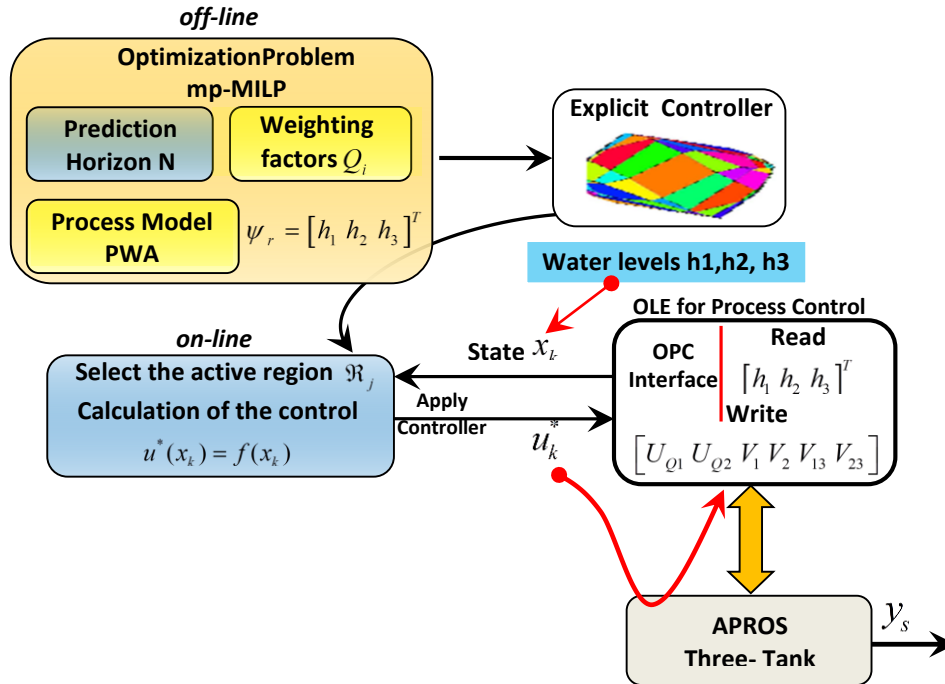


Figure III.7 : Stratégie de la commande hybride hors-ligne appliquée au système à trois-cuves sous l'environnement APROS.

Tableau III.1 : Représentation compacte des tables de recherche contenant la structure de la commande explicite.

Table 1	Table 2	Table 3	Évaluation de la commande en $x_k$
$\mathcal{R}_1$	$F_1$	$G_1$	$F_1 x_k + G_1$
...	...	...	...
$\mathcal{R}_{N_{\mathcal{R}}}$	$F_{N_{\mathcal{R}}}$	$G_{N_{\mathcal{R}}}$	$F_{N_{\mathcal{R}}} x_k + G_{N_{\mathcal{R}}}$

### III.6 Commande prédictive des systèmes hybrides par approche MLD multi-modèle :

Ce paragraphe examine une formulation modifiée du modèle MLD, mise en œuvre pour diminuer le temps de calcul de façon conséquente. On se limite au cas où le système se caractérise par un état continu uniquement. Il n’y a pas de dynamique discrète. Le domaine globale de l’espace d’état est alors divisé en régions distinctes, à l’intérieure desquelles seules les variables actives dans ces régions sont considérées. Ceci mène, pour chaque région, à la conception d’un modèle MLD de taille réduite et en conséquence à des problèmes *MIQP* de complexité limitée [1][4].

L’espace d’état continu est donc divisé en domaines pour lesquels un sous-ensemble de frontières définissant l’interface continu/discret n’est pas franchi, et par conséquent pour lesquels les variables binaires correspondantes sont connues et restent à valeur constante sur l’horizon de prédiction.

Cette réduction de la taille du vecteur  $\delta$  inconnu peut également impliquer dans certains cas une réduction de la taille de  $\{u \text{ et } z\}$  donc une nouvelle simplification du modèle. Avec ces modèles

simples, le problème de temps de calcul peut être surmonté, ce qui peut permettre, si nécessaire pour la commande prédictive, une augmentation de l'horizon de prédiction sans poser de problème pour une application en temps réel.

### III.6.1 Domaines caractéristiques :

Pour mettre en évidence l'approche multi-modèle, il est indispensable d'illustrer les relations d'inclusion existant entre ces domaines selon le pas de prédiction et les variables binaires auxiliaires. Ainsi, le figure III.17 présente une situation générale où les horizons de prédiction sont respectivement  $K$  et  $K'$  pour deux sous-ensembles différents  $\Delta'$  et  $\Delta''$ , avec  $K > K'$ . Dans ce cas, on obtient les relations d'inclusions suivantes :  $R_{\Delta',K}$  et  $R_{\Delta',K'}$  et  $R_{\Delta'',K}$  et  $R_{\Delta'',K'}$ , ce qui justifie le fait que, pour un même sous-ensemble  $\Delta$ , le domaine correspondant à un plus grand horizon est inclus dans le domaine lié à un horizon plus petit. Ainsi, pour le cas où  $\Delta'' \subset \Delta'$  et où les variables binaires auxiliaires communes ont les mêmes valeurs, on obtient les inclusions suivantes :  $R_{\Delta',J} \subset R_{\Delta'',J}$  pour toutes les valeurs de  $J$ , puisque le nombre de variables binaires auxiliaires restant inchangées est supérieure pour  $\Delta'$ , donc le domaine  $R_{\Delta',J}$  a plus de contraintes et de limites que le domaine  $R_{\Delta'',J}$ .

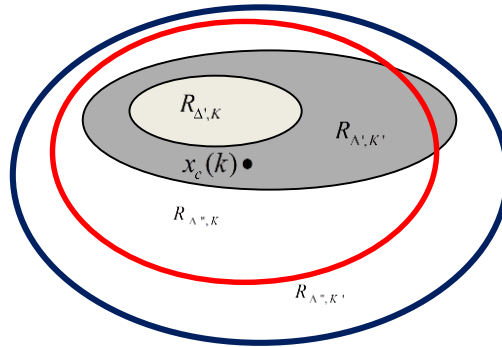


Figure III.8 : Partition de l'espace d'état [1].

### III.6.2 Développement de la stratégie de commande :

Les définitions précédentes nous permettent de construire les sous-régions pour lesquelles des variables auxiliaires binaires ont une valeur fixe. Dès lors, un modèle MLD simplifié peut être développé dans chacune de ces sous-régions. Ainsi, les dynamiques dans ces sous-régions sont exprimées par le système suivant :

$$\begin{cases} x(k+1) = A_i x(k) + B_{i1} u_i(k) + B_{i2} \delta_i(k) + B_{i3} z_i(k) \\ y(k) = C_i x(k) + D_{i1} u_i(k) + D_{i2} \delta_i(k) + D_{i3} z_i(k) \\ E_{i2} \delta_i(k) + E_{i3} z_i(k) \leq E_{i1} u_i(k) + E_{i4} x(k) + E_{i5} \end{cases} \quad (\text{III.26})$$

Où  $i$  représente l'indice du modèle de la région considéré,  $\delta_i$  se compose de variables auxiliaires binaires qui ne font partie de  $\Delta'_{\alpha,j}$  et  $z_i \in \mathfrak{R}^{r_{ic}}$  avec  $r_{ic} \leq r_c$ , et  $u_i \in \mathfrak{R}^{m_{ic}} \times \{0,1\}^{m_{il}}$ , où  $m_{ic} \leq m_c$ ,  $m_{il} \leq m_l$ . De ce fait, chaque modèle MLD simplifié est associé à un ensemble de contraintes définies par  $\Delta$ .



### III.6.3 Découpage en sous-régions pour des modèles MLD multiples :

Afin de mettre en application cette stratégie sur notre système à trois-cuves, l'approche adoptée précédemment nous conduit à définir un ensemble de régions de  $X$  (l'espace de travail), avec  $X = [0-0.62] \times [0-0.62] \times [0-0.62] (m^3)$ .

Une première région de l'espace  $M_1$  peut être définie en considérant que les niveaux du liquide dans les trois réservoirs sont toujours inférieurs à la valeur  $h_v$ , impliquant alors  $\Delta = \delta = [0\ 0\ 0]^T$ . Dans cette région les deux vannes  $V_1$  et  $V_2$  ne sont actives. Par conséquent, les variables auxiliaires continues  $\{z_{0i}\}_{i=1,2,3}$  et  $\{z_i\}_{i=1,2}$  liées aux débits traversant les tubes supérieurs sont inutiles. Il résulte de ces considérations un modèle MLD simplifié pour cette région  $\mathfrak{R}_1$ , faisant intervenir les variables suivants (aucune variable auxiliaires binaires, deux variables auxiliaires continues et seulement deux variables binaires commande) :

$$M_1 : \begin{cases} x = [h_1\ h_2\ h_3]^T \\ u_1 = [Q_1\ Q_2\ V_{13}\ V_{23}]^T \\ z_1 = [z_{13}\ z_{23}]^T \\ \delta_1 = [ \ ]^T \end{cases} \quad (\text{III.27})$$

Une deuxième région se caractérise par les variables auxiliaires binaires  $\delta_2 = [*\ * \ 0]^T$ , des commutations peuvent alors se produire, lorsque les niveaux dans les réservoirs T1 et T2 dépassent le niveau  $h_v$ . Puisque le niveau dans le réservoir R3 est toujours inférieur à  $h_v$ , alors  $z_i(k) = V_i(k)(h_i(k) - h_3(k))$ , pour  $i = 1, 2$  car la direction des débits traversant les conduites sera toujours dirigé vers le troisième réservoir, les  $\{z_{0i}\}_{i=1,2,3}$  sont alors inutiles.

$$M_2 : \begin{cases} x = [h_1\ h_2\ h_3]^T \\ u_2 = [Q_1\ Q_2\ V_1\ V_2\ V_{13}\ V_{23}]^T \\ z_2 = [z_1\ z_2\ z_{13}\ z_{23}]^T \\ \delta_2 = [\delta_{01}\ \delta_{02}]^T \end{cases} \quad (\text{III.28})$$

De la même façon, une troisième région est caractérisée par les variables auxiliaires binaires  $\delta_3 = [1\ 1\ 0]^T$  de sorte que le modèle  $M_3$  dans cette région ne fera intervenir que les variables suivantes :

$$M_3 : \begin{cases} x = [h_1\ h_2\ h_3]^T \\ u_3 = [Q_1\ Q_2\ V_1\ V_2\ V_{13}\ V_{23}]^T \\ z_3 = [z_1\ z_2\ z_{13}\ z_{23}]^T \\ \delta_3 = [ \ ]^T \end{cases} \quad (\text{III.29})$$

Soit particulièrement aucune variable auxiliaire binaire et 4 variable auxiliaires continues.

Une quatrième région se caractérise par  $\delta_4 = [**1]^T$ , des commutations peuvent en effet se produire dans les réservoirs 1 et 2, pendant que le niveau dans le réservoir T3 est toujours supérieur au niveau  $h_1$ . Le quatrième modèle MLD  $M_4$  est donc caractérisé par les variables :

$$M_4 : \begin{cases} x = [h_1 \ h_2 \ h_3]^T \\ u_4 = [Q_1 \ Q_2 \ V_1 \ V_2 \ V_{13} \ V_{23}]^T \\ z_4 = [z_{01} \ z_{02} \ z_1 \ z_2 \ z_{13} \ z_{23}]^T \\ \delta_4 = [\delta_{01} \ \delta_{02}]^T \end{cases} \quad (III.30)$$

Dans cette partie, il y a deux (2) variables auxiliaires binaires et quatre (4) variables auxiliaires continues.

La stratégie de commande appliquée au système à trois cuves sous l'environnement APROS est illustrée par le schéma de la figure III.18 ci-dessous.

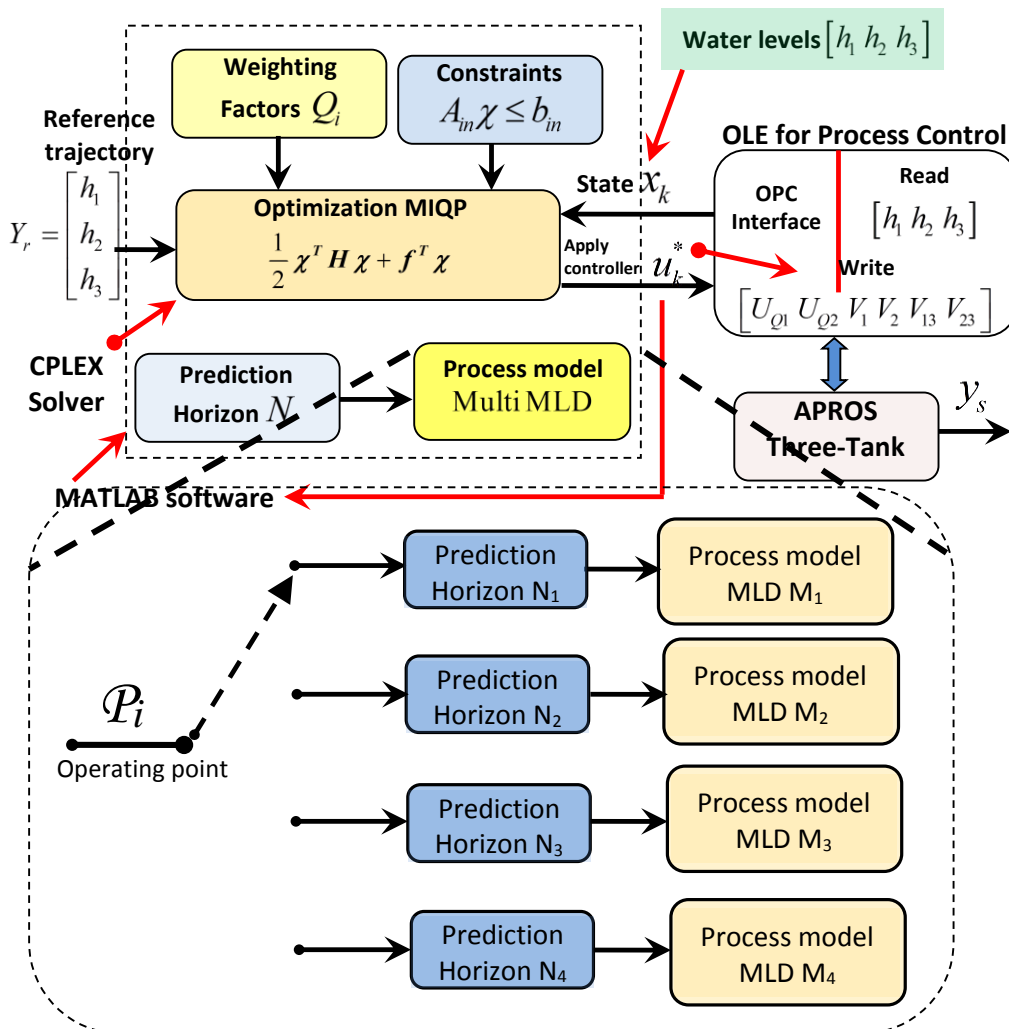


Figure III.9 : Stratégie de la commande hybride multi-modèle en ligne appliquée au système à trois-cuves sous APROS.

**Conclusion :**

La représentation dite MLD d'un système hybride s'avère particulièrement bien adaptée à la mise en œuvre de stratégie de commande prédictive de ces systèmes. Malgré tout, l'explosion combinatoire résultant d'un nombre important de variables binaires issues du formalisme MLD restreint les possibilités d'application temps réel de cette structure pour des systèmes à faible période d'échantillonnage. Ce chapitre a présenté trois stratégies de commande en prenant en compte l'aspect de calcul pendant la phase d'optimisation. L'étude de l'aspect temps de calcul vise à réduire la complexité induite dans la phase d'optimisation et le nombre de problèmes MIQP à résoudre.

En effet, pour deux d'entre-elles, l'objectif est de réduire le nombre de variables d'optimisation binaires, permettant ainsi une moindre complexité des modèles considérés et/ou impliquant une diminution du nombre de problèmes **QP** explicitement considérés. Dans ces deux cas, on effectue une partition de l'espace d'état, les sous-régions étant telles que l'on y garantisse l'absence de certaines commutations sur l'horizon de prédiction. Pour la deuxième stratégie, l'objectif est de réaliser l'optimisation hors ligne et le calcul de la commande revient à une simple opération d'appartenance à une région polyédrale donnée avec retour d'état.

L'application de ces trois structures au même banc d'essais à trois cuves sous l'environnement APROS, dans les mêmes conditions, apporte des points de comparaison très intéressants pour la mise en œuvre d'une stratégie de commande prédictive sur les systèmes hybrides sous formes MLD « modifiées », implantables de façon réaliste en temps réel.

# *Chapitre IV*

## Application & Simulation de la commande du système hybride à trois-cuves

### IV.1 Introduction :

Dans ce chapitre nous présentons deux algorithmes de commandes des systèmes dynamiques hybrides appliqués sur un benchmark COSY (three tank system) composés par trois réservoirs interconnectés par un jeu des vannes TOR et alimentés par des pompes. L'étude effectuée à travers les chapitres précédents ont permis de définir l'ensemble des concepts mettant en œuvre un formalisme global de modélisation et de commande à horizon glissant des systèmes hybrides. En effet, il a été montré que la représentation « MLD » d'un système hybride rendait possible l'implémentation simple d'algorithmes de commande prédictive. Pour cela des variables auxiliaires binaires et continues modélisant les interfaces entre les parties discrets et continues ainsi que les opérations logiques sont ajoutées. Le problème de commande est converti vers un problème d'optimisation dans le contexte de la programmation linéaire mixte avec l'utilisation de l'algorithme « Branch & Bound ». Cette stratégie de commande est limitée par le temps de calcul pendant la phase d'optimisation, qui augmente exponentiellement avec le nombre de variables.

La technique de la programmation multiparamétrique permet d'obtenir une solution optimale comme une fonction des paramètres du système. Le problème de commande optimale de système modélisé sous forme PWA a ainsi présenté comme un problème de programmation multiparamétrique (mp-MILP), les entrées et les états du système  $y$  sont traités respectivement comme des variables d'optimisation et des paramètres. L'approche multiparamétrique fournit, par un calcul hors-ligne, un ensemble complet de lois de commande optimales du système en fonction des variables d'état, ainsi que les régions dans l'espace d'état où ces fonctions restent optimales. Il est montré que ces lois sont linéaires et les régions correspondantes sont définies par des inégalités linéaires.

### IV.2 Description du banc d'essai « trois cuves » :

Le banc d'essai représenté par la figure V.1 est en fait un banc destiné pour des problèmes de commande et de détection de pannes de systèmes hybrides. Le système a été développé au sein du programme de recherche « Control of Complex Systems » (COSY) [7][8].

Le système se compose de trois réservoirs alimentés par deux pompes indépendantes de débit  $Q_1$  et  $Q_2$  dans les réservoirs 1 et 2 respectivement. Ces deux pompes sont manœuvrées de manière continue de 0 jusqu'à un débit maximum  $Q_{\max}$ . Les quatre vannes  $V_1, V_2, V_{13}$  et  $V_{23}$  commandent les débits entre les réservoirs comme est montré par la figure V.1. Ces quatre vannes sont de type tout ou rien (ouverte si  $V_i = 1$ , fermée sinon). La vanne  $V_{N_3}$  commande le débit nominal de sortie du réservoir central.

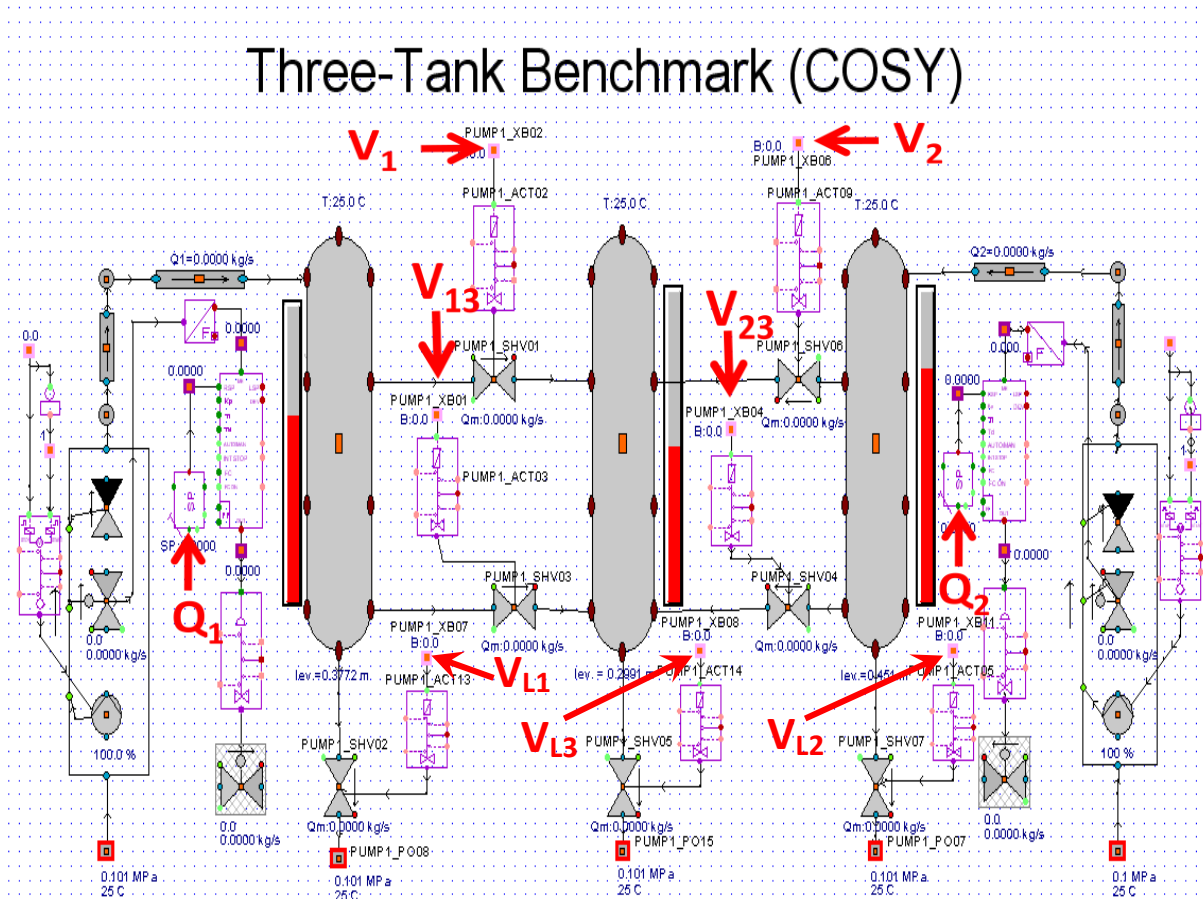


Figure IV.1 : Modèle APROS du système à trois-cuves « COSY ».

La modélisation analytique a été abordée au chapitre II (équations du (II.1) à (II.12)). Ces expressions permettent alors l'élaboration d'un modèle MLD du benchmark.

Ces expressions permettent alors l'élaboration d'un modèle MLD.

### IV.3 Mise sous forme MLD :

De ces expressions, un modèle MLD peut être obtenu par les techniques classiques exposées dans [1][3][4][30] en introduisant les variables auxiliaires continues et binaires requises par la traduction des interfaces, de sorte que l'on fera intervenir les vecteurs suivants :

$$x = [h_1 \ h_2 \ h_3]^T \quad (IV.1)$$

$$u = [Q_1 \ Q_2 \ V_1 \ V_2 \ V_{13} \ V_{23} \ V_{L1} \ V_{L2} \ V_{N3}]^T \quad (IV.2)$$

$$\delta = [\delta_{01} \ \delta_{02} \ \delta_{03}]^T \quad (IV.3)$$

$$z = [z_{01} \ z_{02} \ z_{03} \ z_1 \ z_2 \ z_{13} \ z_{23} \ z_{L1} \ z_{L2} \ z_{N3}]^T \quad (IV.4)$$

Avec :

$$[\delta_{0i}(k) = 1] \leftrightarrow [h_i(k) \geq h_v] \quad i=1,2,3 \quad (IV.5)$$

$$z_{0i}(k) = \delta_{0i}(k)(h_i(k) - h_v) \quad i=1,2,3 \quad (IV.6)$$

$$z_i(k) = V_i(k)(z_{0i}(k) - z_{03}(k)) \quad i=1,2 \quad (IV.7)$$

$$z_{i3}(k) = V_{i3}(k)(h_i(k) - h_3(k)) \quad i=1,2 \quad (IV.8)$$

A partir de ces relations et en utilisant la technique de discrétisation au premier ordre (Euler) :

$\dot{h} \rightarrow \frac{h(k+1) - h(k)}{T_s}$ , on obtient la forme discrète suivante :

$$\begin{cases} h_1(k+1) = h_1(k) + \frac{1}{A}(Q_1 - k_1 z_1(k) - k_{13} z_{13}(k) - k_{L1} z_{L1}(k)) \\ h_2(k+1) = h_2(k) + \frac{1}{A}(Q_2 - k_1 z_2(k) - k_{23} z_{23}(k) - k_{L2} z_{L2}(k)) \\ h_3(k+1) = h_3(k) + \frac{1}{A}(k_1 z_1(k) + k_{13} z_{13}(k) + k_2 z_2(k) + k_{23} z_{23}(k) - k_{N3} z_{N3}(k)) \end{cases} \quad (IV.9)$$

La structure MLD impose donc l'ajout de trois (03) variables auxiliaires binaires et sept (07) variables auxiliaires continues, on rappelle que le vecteur de commande inclut deux (02) variables continues et quatre (04) variables binaires et que le vecteur d'état ne contient que trois (03) variables continues car le système ne présente pas de dynamique discrète. Le fichier HYSDEL du modèle MLD pour ce système (banc d'essai se trouve en Annexe A).

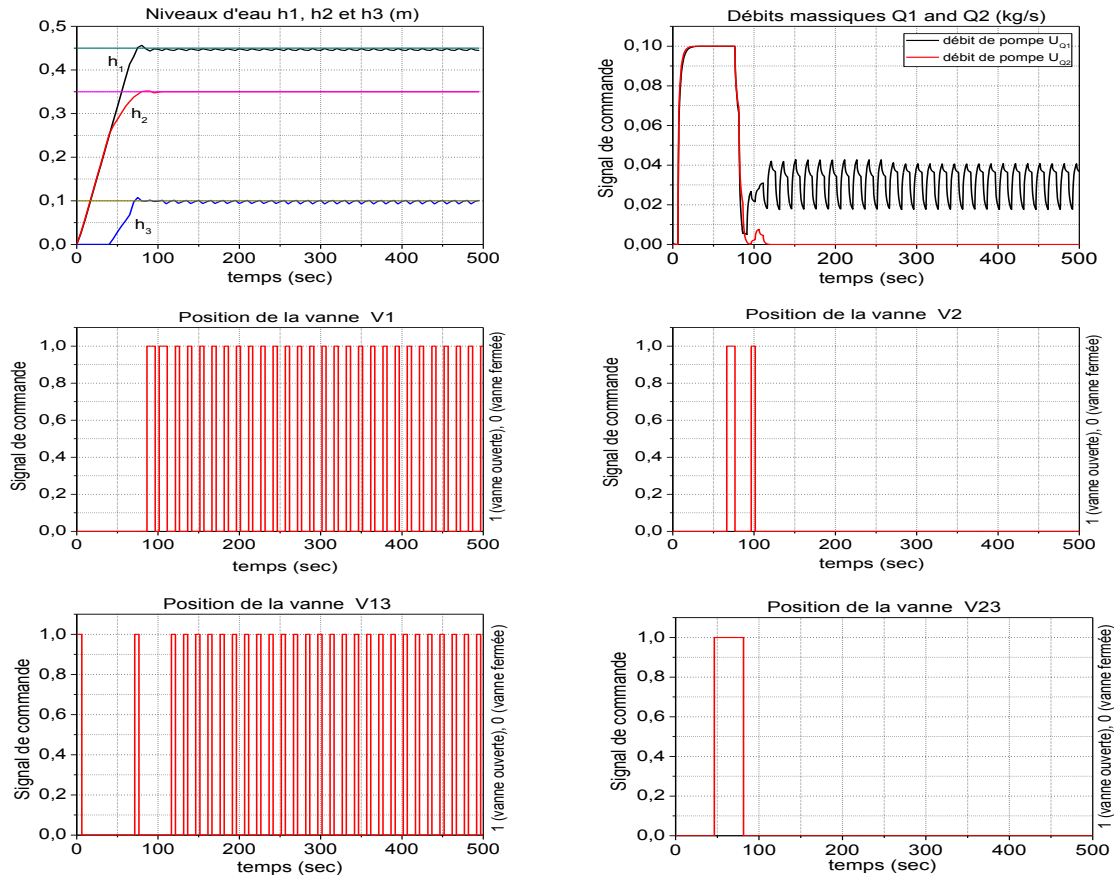
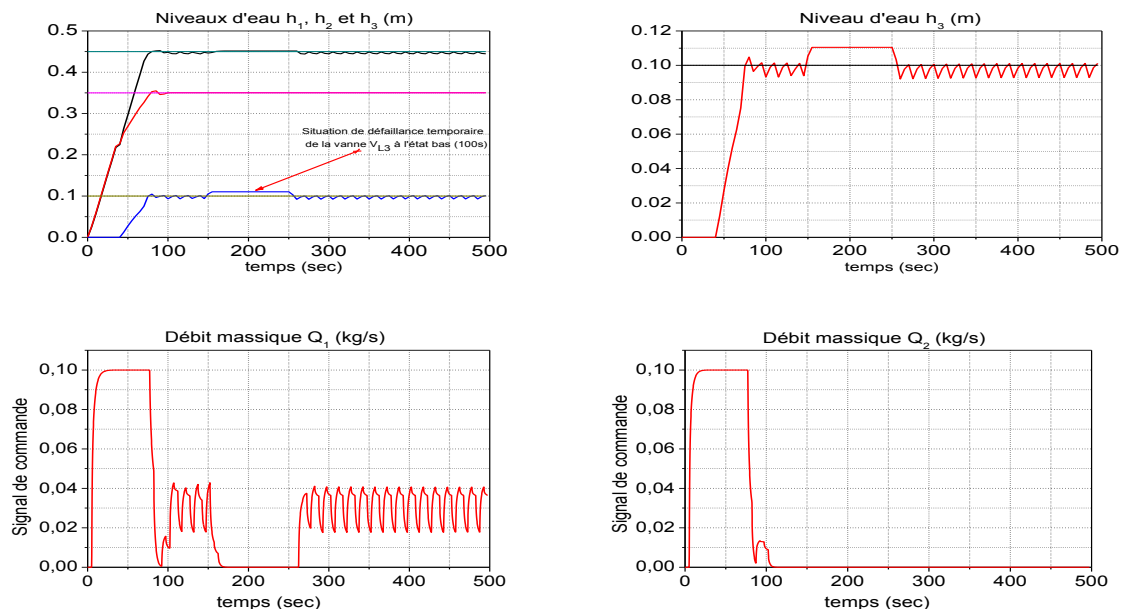


Figure IV.2 Commande prédictive des niveaux  $h_1$ ,  $h_2$  et  $h_3$   
(Horizon de prédiction  $N=3$ ,  $Q_u=610^3$ ,  $Q_d=Q_z=10^{-2}$ , vanne  $V_{L3}$  (=1) est maintenue ouverte)

L'approche hybride MLD a été appliquée dans un premier temps sur le modèle APROS à trois-cuves avec la configuration suivante : le vecteur commande  $u = [Q_1 \ Q_2 \ V_1 \ V_2 \ V_{13} \ V_{23}]^T$  avec la vanne de sortie sur le réservoir R3  $V_{N_3}$  est maintenue complètement ouverte, par contre les deux vannes de sortie  $V_{L_1}$  et  $V_{L_2}$  sur les réservoirs R1 et R2 sont complètement fermées (voir la figure IV.2). Le modèle MLD élaboré pour la synthèse de la commande MPC est composé par 2 variables continues, 4 variables binaires, 3 variables binaires auxiliaires et 7 variables continues auxiliaires avec 44 contraintes linéaires mixtes entières. Le problème de commande MPC est transformé en problème d'optimisation linéaire avec contraintes mixtes entières donnant sur un horizon de prédiction choisi égal à celui de commande  $N=3$ , à 48 variables continues et binaires avec 132 contraintes linéaires mixtes entières. Les commandes calculées par programme d'optimisation **MIQP** avec un solveur **CPLEX** sont appliquées via l'interface **OPC** (OLE for Control Process) sur le modèle pour atteindre les spécifications de niveau d'eau  $h_1=0.45\text{m}$  ;  $h_2=0.35$  et  $h_3=0.1$  m. Les trois réservoirs étant initialement vides (niveau=0). Notons que le niveau du troisième réservoir présente des oscillations autour de la référence, car le niveau  $h_3=0.1$  m ne correspond pas à un point d'équilibre : pour ces consignes de niveau, le débit sortant du troisième réservoir  $Q_{N_3}$  n'est égal à aucune combinaison des débits entrants ( $Q_{13V_1}$ ,  $Q_{23V_2}$ ,  $Q_{13V_{13}}$  et  $Q_{23V_{23}}$ ). Le niveau  $h_2$  ne présente pas des oscillations et que les deux vannes  $[V_2 \ V_{23}]$  sont maintenues fermées après avoir atteint la consigne  $h_{r2}=0.35\text{m}$ . Par contre, le niveau sur le réservoir R1 présente des oscillations induites par l'ouverture et la fermeture des deux vannes  $[V_1 \ V_{13}]$  pour le réglage du niveau  $h_1$  et  $h_3$  autour de leurs consignes. Pour le choix des pondérations sur le critère d'optimisation, notre choix est particulièrement destiné à la pénalisation de l'erreur de poursuite entre les niveaux et leurs consignes affectées par la matrice  $Q_y=6000$ .





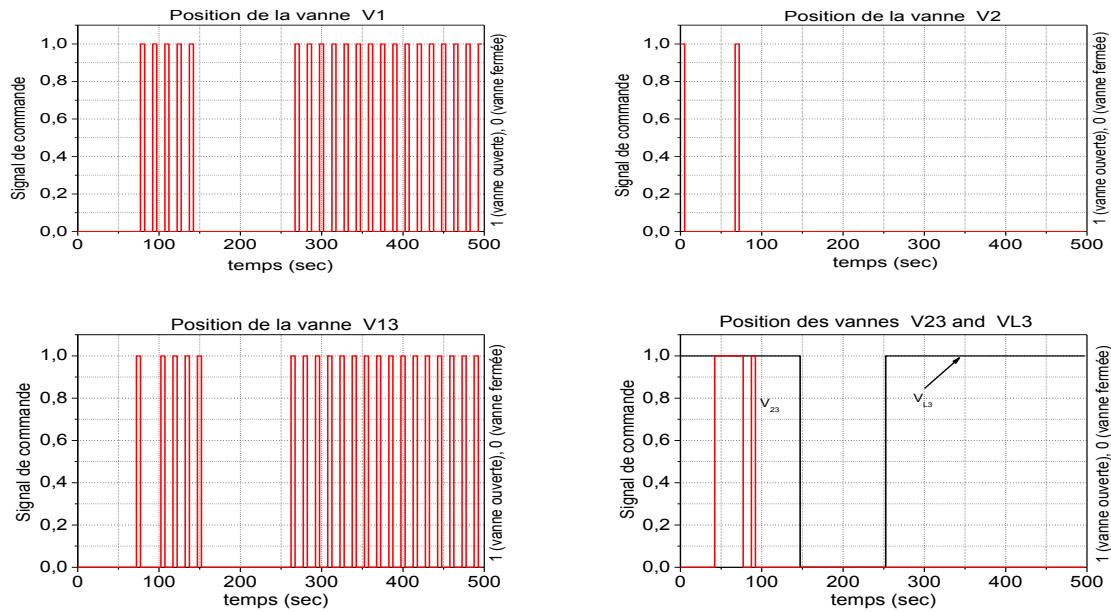
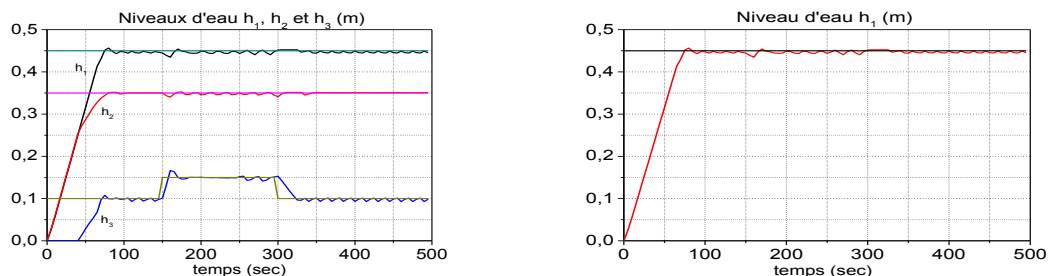


Figure IV.3 Commande prédictive des niveaux  $h_1$ ,  $h_2$  et  $h_3$   
 (Horizon de prédiction  $N=3$ ,  $Q_u=610^3$ ,  $Q_d=Q_z=10^{-2}$ , vanne  $V_{L3}$  (=1) est maintenue ouverte avec situation de blocage à l'état bas pendant 100 secondes)

Dans le but de tester la robustesse de notre algorithme de commande vis-à-vis d'une situation de blocage de la vanne de sortie VL3 sur le troisième réservoir R3. La même configuration précédente est réutilisée avec la situation de blocage suivante : à l'instant  $t=150$  secondes, la vanne VL3 est maintenue fermée pendant 100 secondes (fin du blocage à l'instant 250 s). Le régulateur prédictif MPC réagit rapidement à cette situation après une seule période d'échantillonnage avec un dépassement inférieure à 10% de la valeur de la consigne  $h_3$ . Cette perturbation sur le fonctionnement du système est rejetée par le régulateur en agissant sur l'état des deux vannes  $[V_1 \ V_{13}]$  en les conduisant vers la fermeture et le débit  $Q_I$  à l'arrêt total (voir la figure IV.3). A partir de l'instant  $t=250$  secondes, le système reprend son fonctionnement correctement en agissant sur les vannes d'actions entre le réservoir R1 et R3 en procédant au dé-remplissage de ce dernier et les oscillations autour du niveau  $h_1$  et  $h_3$  apparaissent de nouveau pour maintenir le réglage autour de leurs consignes de références.



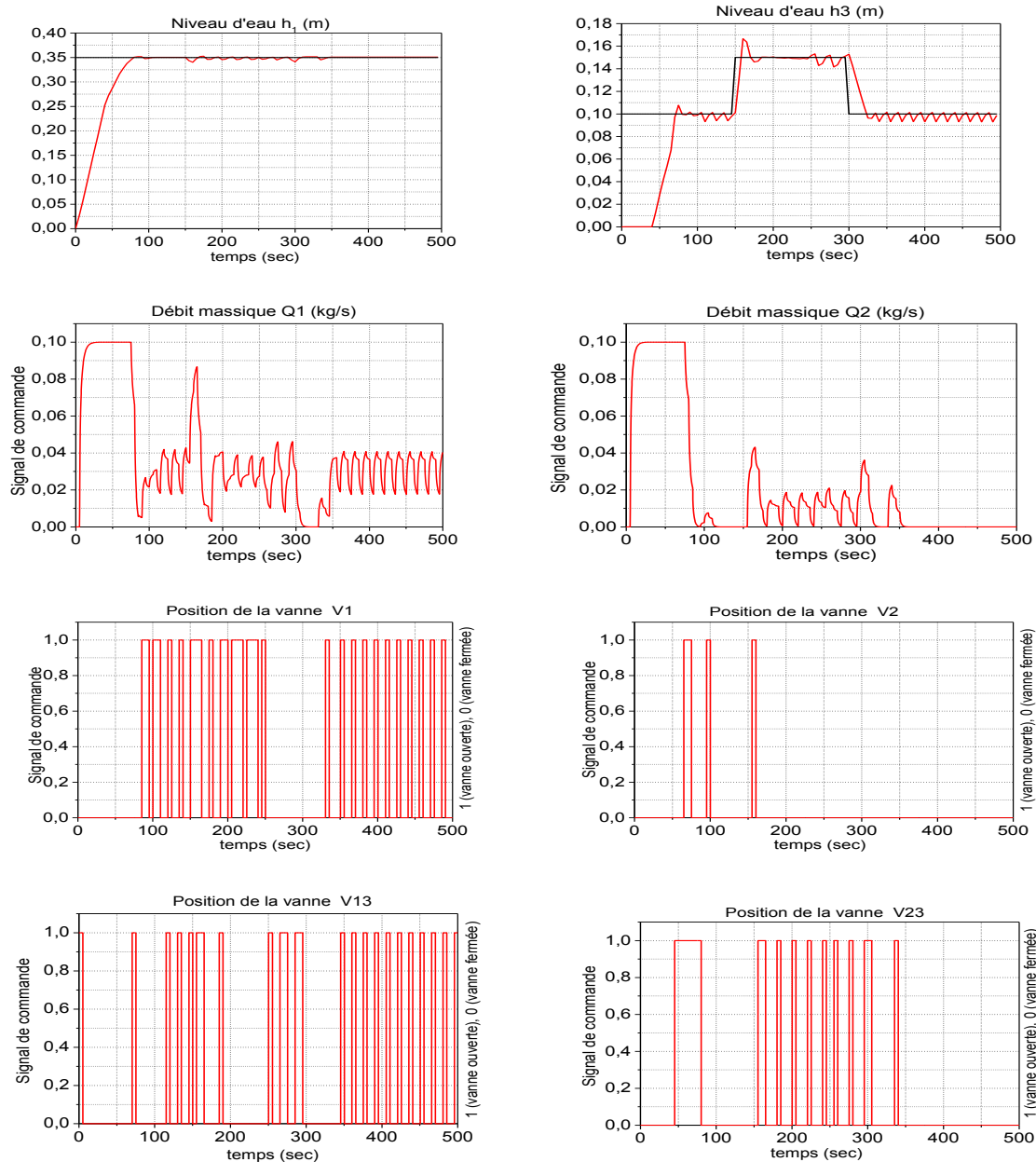


Figure IV.4 Commande prédictive des niveaux  $h_1$ ,  $h_2$  et  $h_3$   
 (Horizon de prédiction  $N=3$ ,  $Q_u=610^3$ ,  $Q_d=Q_z=10^{-2}$ , vanne  $V_{L3}$  (=1) est maintenue ouverte avec variation de consigne du niveau  $h_3$ )

La figure IV.4 montre une situation de simulation dans laquelle un changement de la valeur de consigne sur le niveau  $h_3$  est opéré pendant une période de temps de 150 secondes avec une augmentation de 50% de sa valeur courante qui est égale à 0,15m. La première lecture des résultats de simulations nous permet de constater que la stratégie de commande prédictive anticipe les variations de la référence, ce qui est un des avantages de la commande prédictive et que la variation de niveau du troisième réservoir R3 de 0,15 à 0,10 m prend plus de temps que la variation de 0,1 à 0,15 m à cause des contraintes physiques du benchmark. Une seconde remarque sur les variations apparentes sur la réponse du système est le dépassement sur le niveau  $h_3$  avec un taux moins de 10% sur la valeur de consigne. Cette situation fait intervenir

toutes les variables de commandes continues et binaires  $u = [Q_1 \ Q_2 \ V_1 \ V_2 \ V_{13} \ V_{23}]^T$  pendant la transition du niveau  $h_3$  avec des petites oscillations sur les trois niveaux de liquides  $h_1$ ,  $h_2$  et  $h_3$ . Après cette transition, les oscillations sur le deuxième niveau  $h_2$  disparaissent définitivement par l'action sur les deux vannes  $[V_2 \ V_{23}]$  et le débit  $Q_2$  par annulation de leurs contributions au réglage du niveau  $h_3$ .

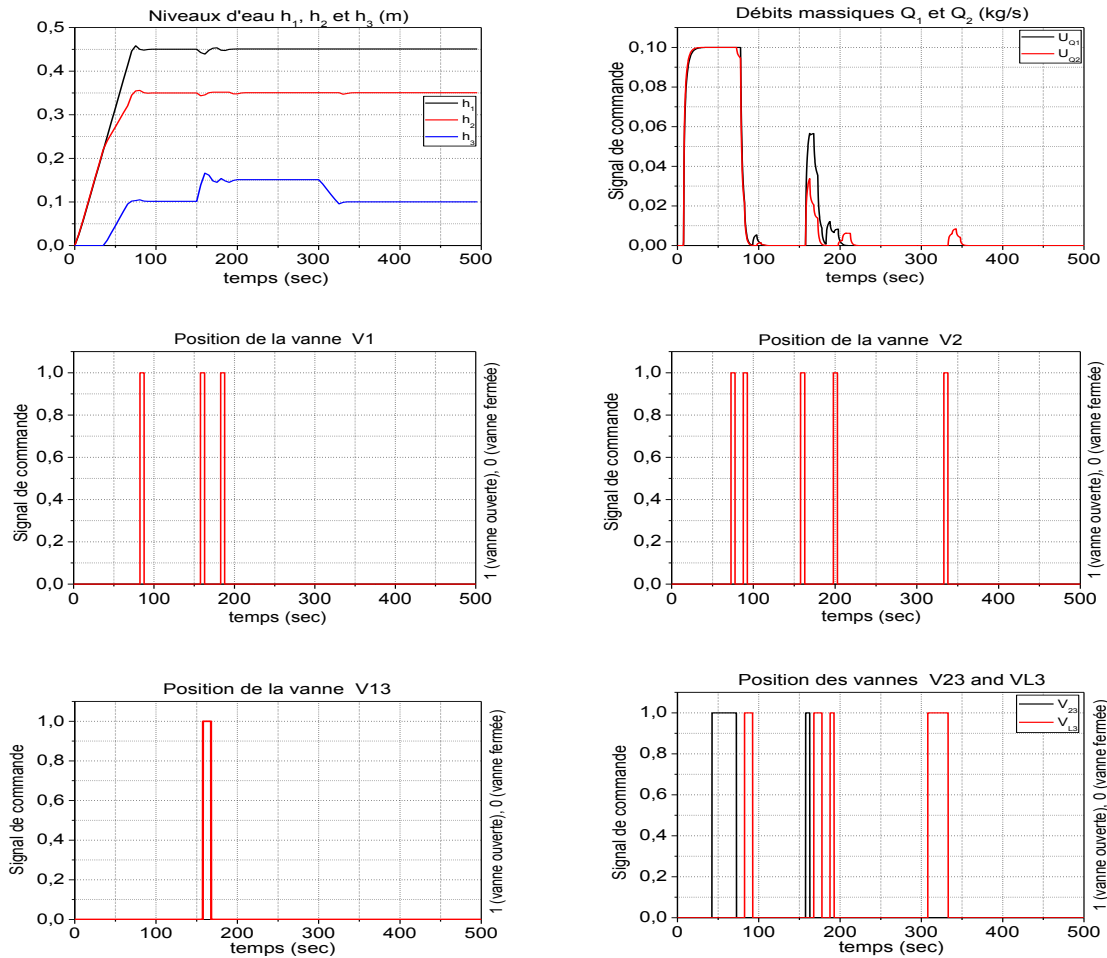


Figure IV.5 : Commande prédictive des niveaux  $h_1$ ,  $h_2$  et  $h_3$  (Horizon de prédiction  $N=3$ ,  $Q_u=610^3$ ,  $Q_d=Q_z=10^{-2}$ , vanne  $V_{L3}$  est contrôlée avec variation de consigne du niveau  $h_3$ )

La lecture des résultats précédents par la présence des oscillations après réglage, nous mène dans ce cas à apporter une modification sur le vecteur de commande en impliquant la vanne de sortie du réservoir R3 dans la stratégie de régulateur prédictif MPC. Dans cette nouvelle configuration, le vecteur de commande devient  $u = [Q_1 \ Q_2 \ V_1 \ V_2 \ V_{13} \ V_{23} \ V_{N3}]^T$ . Le modèle MLD élaboré pour la synthèse de la commande MPC est composé par 2 variables continues, 5 variables binaires, 3 variables binaires auxiliaires et 8 variables continues auxiliaires avec 48 contraintes linéaires mixtes entières. Le problème de commande MPC est transformé en un problème d'optimisation quadratique avec contraintes mixtes entières donnant sur un horizon de prédiction choisi égal à celui de commande  $N=3$ , à 54 variables continues et binaires avec 144 contraintes linéaires mixte entières. La réponse du benchmark illustrée par la figure IV.5,

montrent bien l'efficacité du régulateur prédictif pour l'élimination des petites oscillations apparente sur le régime stationnaire sans affectée le temps du transitoire pour le niveau  $h_3$ . Une augmentation du modèle MLD, avec implication d'autres variables de commande permet d'améliorer la réponse du système, seulement avec un volume de calcul important en temps réel pour la résolution problème quadratique linéaire mixte entier **MIQP avec contraintes**.

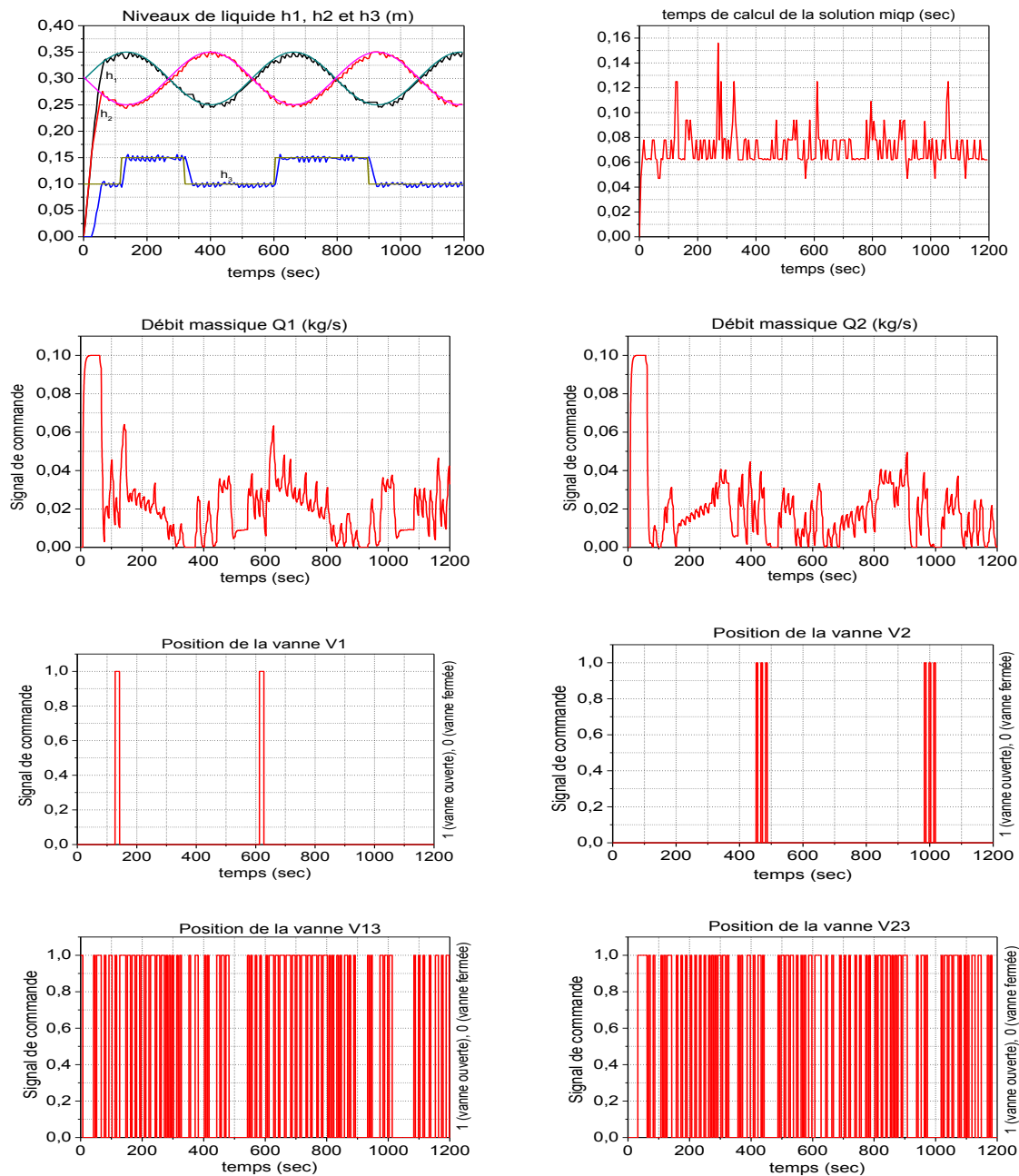


Figure IV.6 : Commande prédictive des niveaux  $h_1$ ,  $h_2$  et  $h_3$   
(Horizon de prédiction  $N=3$ ,  $Q_u=610^3$ ,  $Q_d=Q_z=10^{-2}$ , vanne  $V_{L3}$  est maintenue ouverte avec variation sinusoïdale des consignes des niveaux  $h_1$  et  $h_2$ )

La robustesse du régulateur prédictif visa vis une référence variable de forme sinus sur les niveaux  $h_1$  et  $h_2$  avec une variation en échelon sur le niveau  $h_3$  est effectuée avec la configuration initiale (voir la figure IV.6). Le système répond parfaitement à la commande

MPC hybride par le vecteur  $u = [Q_1 \ Q_2 \ V_1 \ V_2 \ V_{13} \ V_{23}]^T$  appliqué sur le benchmark APROS. Les trois niveaux  $h_1, h_2$  et  $h_3$  poursuivent leurs consignes en forme et en temps. On constate néanmoins, des oscillations sur les niveaux avec un effort un peu excitant sur les débits d'alimentations des deux réservoirs R1 et R2, ainsi des commutations assit importante sur les deux vannes  $[V_{13} \ V_{23}]$  pour asservir les niveaux de liquides.

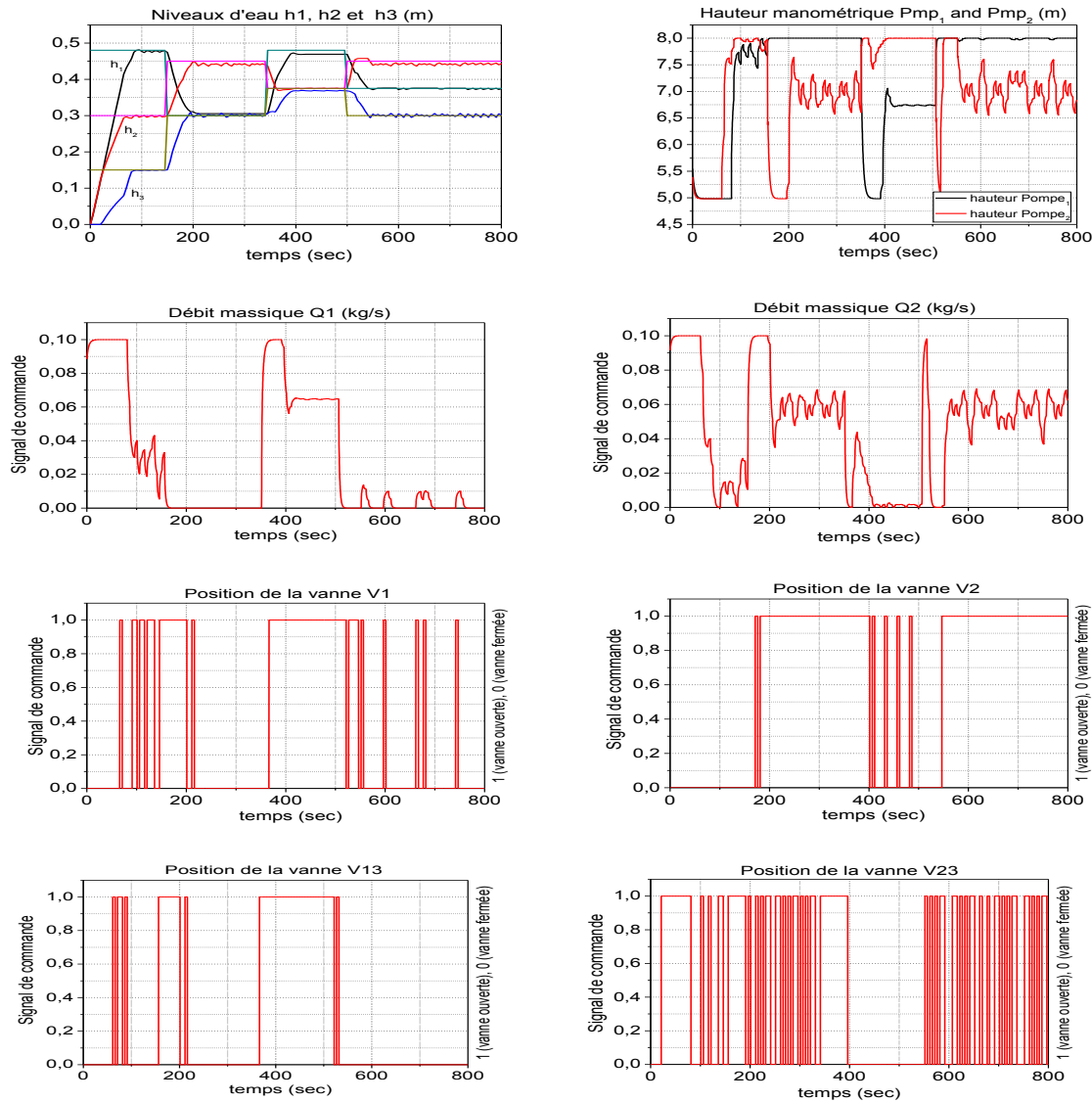


Figure IV.7 : Commande prédictive des niveaux  $h_1, h_2$  et  $h_3$   
(Horizon de prédiction  $N=3$ ,  $Q_u=610^3$ ,  $Q_d=Q_z=10^{-2}$ , vanne  $V_{L3}$  est maintenue ouverte avec variation en échelon des consignes des niveaux  $h_1, h_2$  et  $h_3$ )

Cette partie sera consacrée à l'évaluation du vecteur de commande sur la réponse transitoire et stationnaire avec des consignes en échelons variables pour les trois niveaux de liquides. Les résultats de simulations affichées par la figure IV.7 pour une poursuite de trajectoire comme suit : les niveaux des liquides  $[h_1 h_2 h_3]$  sont régis par les consignes  $[hr_1 \ hr_2 \ hr_3]$  qui passe du  $[0 \ 0 \ 0]$  à  $[80 \ 50 \ 25]\%$ , ensuite de  $[80 \ 50 \ 25]\%$  à  $[50 \ 75 \ 50]\%$ , après de  $[50 \ 75 \ 50]\%$  à  $[80 \ 62.50 \ 62.50]\%$  et finalement de  $[80 \ 62.50 \ 62.50]\%$  à  $[62.50 \ 75 \ 50]\%$ . Cette configuration de remplissage et de vidage permet de mieux solliciter les modes opératoires des différents

réservoirs. La stratégie de commande prédictive nous assure parfaitement la poursuite de trajectoire dans les différents modes. On constate que pendant le deuxième mode lors du passage de  $[80 \ 50 \ 25]\%$  à  $[50 \ 75 \ 50]\%$ , les deux vannes et le débit  $[Q_1 \ V_1 \ V_{13}]$  du premier réservoir après avoir atteint le niveau stationnaire n'interviennent pas dans le réglage des deux autres réservoirs R2 et R3 avec une sollicitation de la commande du débit  $Q_2$  ainsi que des commutations fréquentes sur la vanne  $V_{23}$ . D'autre part, le passage du mode  $[80 \ 62.50 \ 62.50]\%$  à  $[62.50 \ 75 \ 50]\%$  implique d'avantage le deuxième réservoir R2 du fait qu'il se trouve dans un état de remplissage et que le premier réservoir R1 dans un état de vidage, ce qui explique la présence des oscillations au niveau du deuxième et troisième réservoir au régime stationnaire accompagné d'erreur statique sur le niveau  $h_2$ .

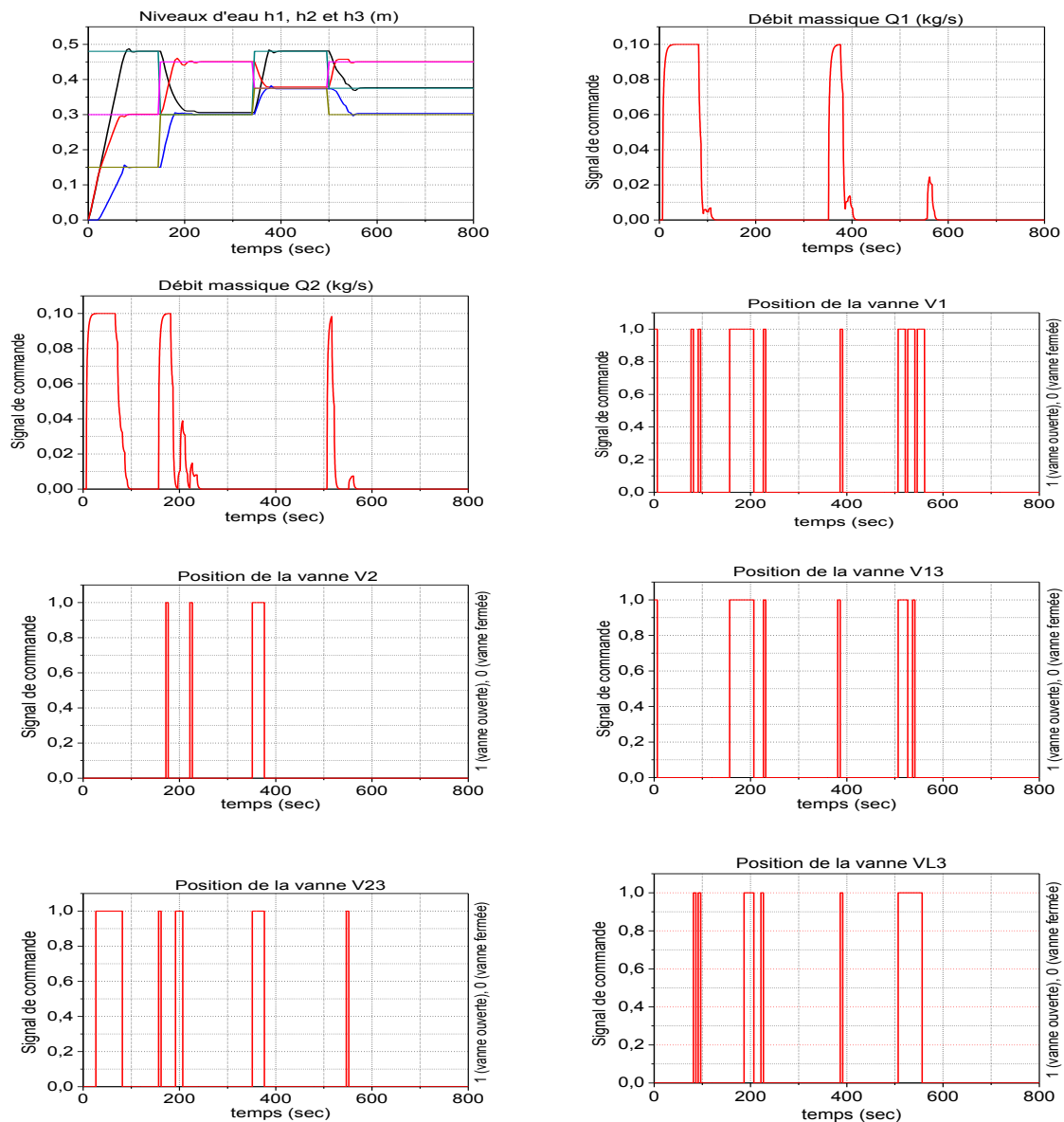
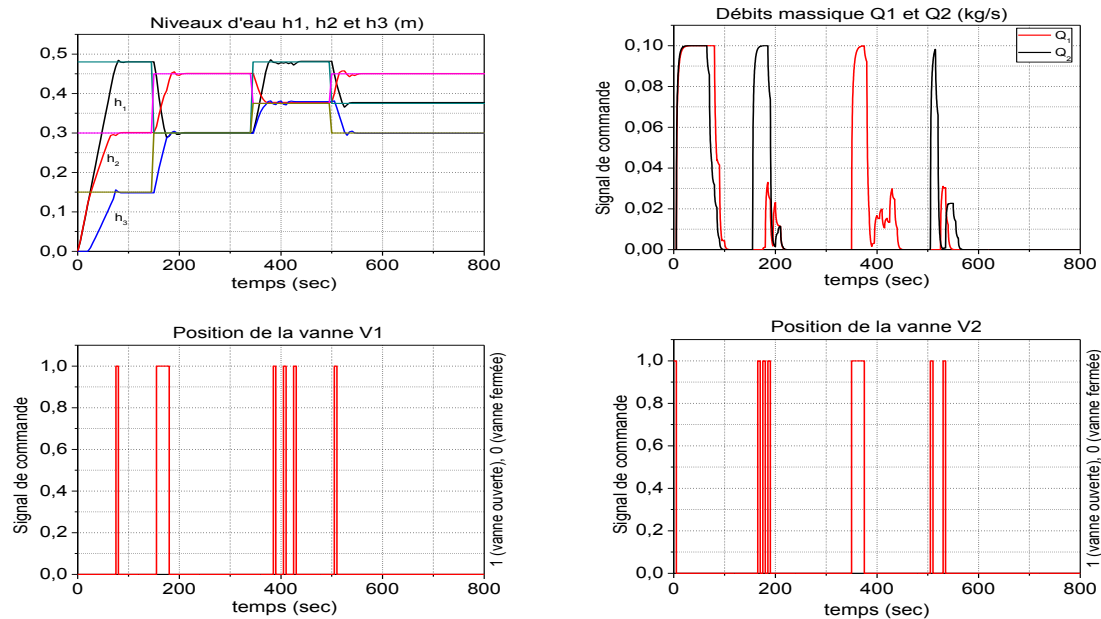


Figure IV.8 : Commande prédictive des niveaux  $h_1$ ,  $h_2$  et  $h_3$   
(Horizon de prédiction  $N=3$ ,  $Q_u=610^3$ ,  $Q_d=Q_z=10^{-2}$ , vanne  $V_{L3}$  est contrôlable avec variation en échelon des consignes des niveaux  $h_1$ ,  $h_2$  et  $h_3$ )

Afin d'ajuster le régulateur prédictif hybride (régime transitoire et stationnaire), on procède dans un premier lieu d'impliquer la vanne de sortie  $V_{L3}$  dans le vecteur de commande  $u = [Q_1 \ Q_2 \ V_1 \ V_2 \ V_{13} \ V_{23} \ V_{N3}]^T$ . On constate que le réglage devient plus fin sans présence des oscillations en régime stationnaire avec des petits dépassements sur les trois niveaux. Les commandes continues étant cette fois-ci sont moins excitantes, avec moins de commutations sur les différentes vannes (voir la figure IV.8). Le régime transitoire de montée et de descente n'est affecté par cette configuration. Ce qui nous amène à faire une deuxième modification sur le vecteur de commande par l'implication de la vanne de contrôle  $V_{L1}$  sur le réservoir R1. Le vecteur de commande dans cette nouvelle configuration devient  $u = [Q_1 \ Q_2 \ V_1 \ V_2 \ V_{13} \ V_{23} \ V_{L1} \ V_{N3}]^T$ . Le modèle MLD élaboré pour la synthèse de la commande MPC est composé par 2 variables continues, 6 variables binaires, 3 variables binaires auxiliaires et 9 variables continues auxiliaires avec 52 contraintes linéaires mixtes entières. Le problème de commande MPC est converti en problème d'optimisation quadratique avec contraintes mixte entières donnant sur un horizon de prédiction choisi égal à celui de commande  $N=3$ , à 60 variables continues et binaires avec 156 contraintes linéaires mixte entières. La réponse du benchmark illustrée par la figure IV.9, montrent bien l'efficacité du régulateur prédictif pour l'illumination des petites oscillations apparente sur le régime stationnaire avec une amélioration du temps transitoire pour les niveaux  $h_1$  et  $h_3$ . Les dépassements observés dans les simulations précédentes ont été réduites et que le transitoire sur le niveau  $h_2$  dans le cas de remplissage ou de vidage n'est pas affecté par cette configuration. Une augmentation du modèle MLD, avec implication d'autres variables de commande permet d'améliorer la réponse du système, seulement avec une charge de calcul en temps réel pour la résolution problème quadratique linéaire mixte entier MIQP avec contraintes.



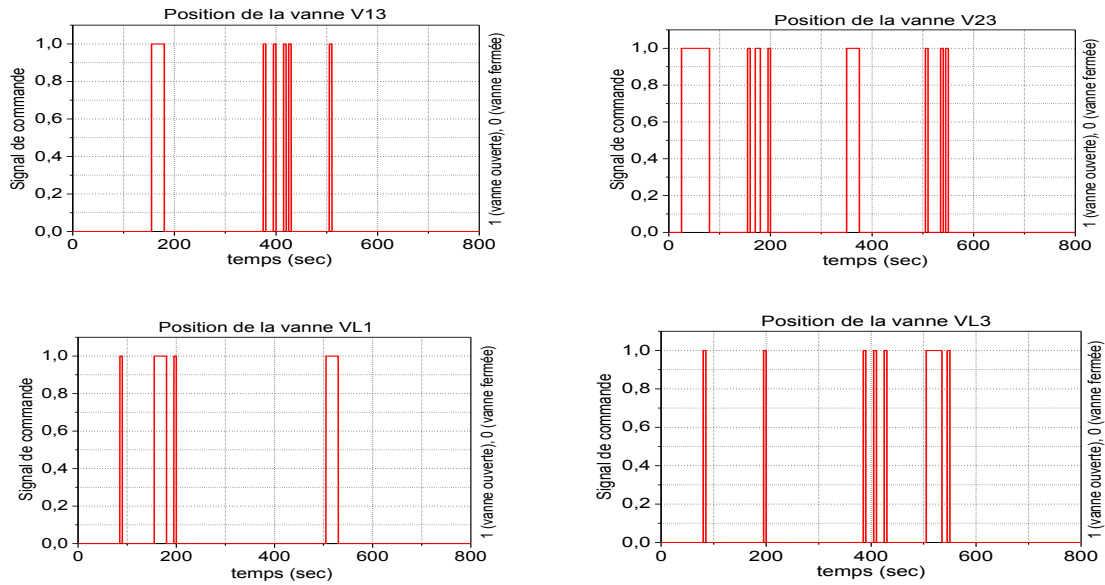
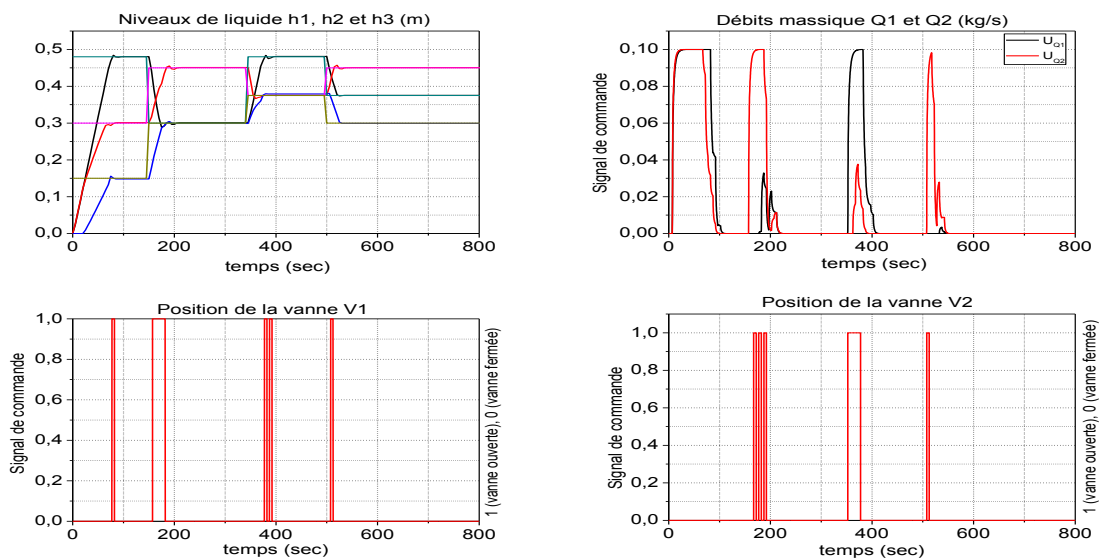


Figure IV.9 : Commande prédictive des niveaux  $h_1$ ,  $h_2$  et  $h_3$   
 (Horizon de prédiction  $N=3$ ,  $Q_u=610^3$ ,  $Q_d=Q_z=10^{-2}$ , vannes  $V_{L1}$  et  $V_{L3}$  sont contrôlées avec variation en échelon des consignes des niveaux  $h_1$ ,  $h_2$  et  $h_3$ )

Dans cette présente simulation, et dans le souci d'agir sur le comportement dynamique de niveau du deuxième réservoir, on ajoute une variable de commande au vecteur de contrôle qui devient  $u = [Q_1 \ Q_2 \ V_1 \ V_2 \ V_{13} \ V_{23} \ V_{L1} \ V_{L2} \ V_{N3}]^T$ . Le variable de commande  $V_{L2}$  agit sur la vanne de sortie du deuxième réservoir R2. Le modèle MLD élaboré pour la synthèse de la commande MPC est composé par 2 variables continues, 7 variables binaires, 3 variables binaires auxiliaires et 10 variables continues auxiliaires avec 56 contraintes linéaires mixtes entières. Le problème de commande MPC est converti en problème d'optimisation quadratique avec contraintes mixte entières donnant sur un horizon de prédiction choisi égal à celui de commande  $N=3$ , à 66 variables continues et binaires avec 168 contraintes linéaires mixte entières. La réponse du benchmark illustrée par la figure IV.9 confirme l'efficacité du réglage hybride MPC pour une poursuite de trajectoire en temps réel.





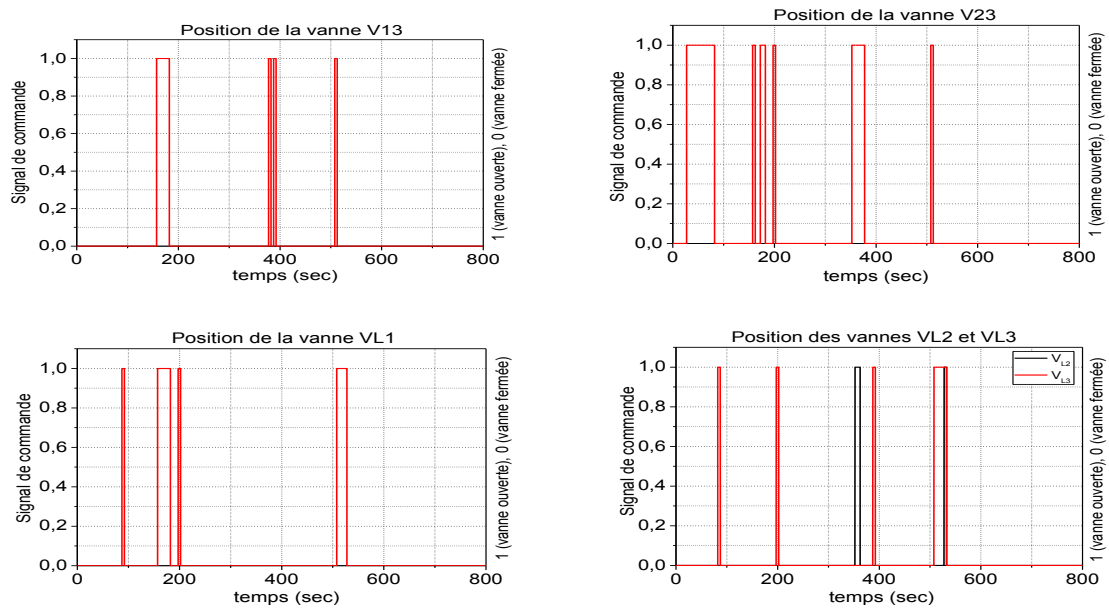
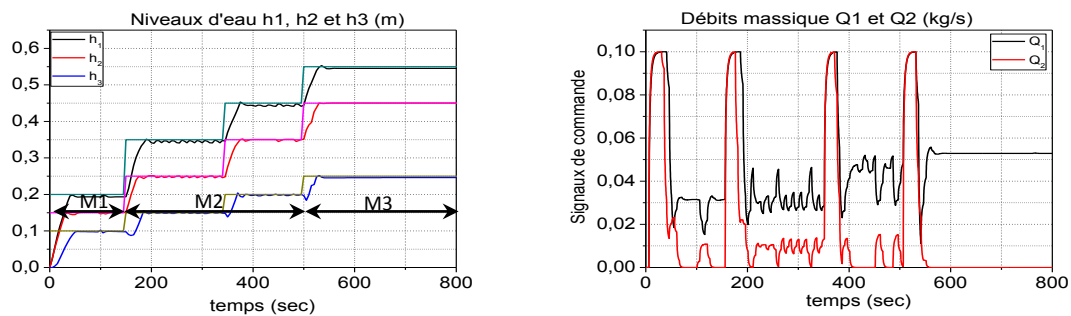


Figure IV.10 : Commande prédictive des niveaux  $h_1$ ,  $h_2$  et  $h_3$   
 (Horizon de prédiction  $N=3$ ,  $Q_u=610^3$ ,  $Q_d=Q_z=10^{-2}$ , vannes  $V_{L1}$ ,  $V_{L2}$  et  $V_{L3}$  sont contrôlées avec variation en échelon des consignes des niveaux  $h_1$ ,  $h_2$  et  $h_3$ )

Enfin, pour terminer avec la commande prédictive hybride **MPC**, on donne les résultats obtenus par simulation de l'application sur un benchmark sur un modèle **APROS** à trois réservoirs de la configuration d'une commande MPC avec Multi-modèle MLD (voir la figure IV.10). La simulation effectuée contient trois modes opératoires **M1**, **M2** et **M3**, chacun de ces modes appartient un espace limité par les contraintes physique du benchmark. Le mode **M1** est choisi dans l'espace de fonctionnement  $[0-0.25] \times [0-0.25] \times [0-0.1](m^3)$ , le mode **M2** est choisi dans l'espace  $[0.25-0.45] \times [0.25-0.45] \times [0-0.20](m^3)$  et le mode **M3** est choisi dans l'espace de travail  $[0.45-0.55] \times [0.45-0.55] \times [0.20-0.30](m^3)$  avec une configuration initiale. Les résultats de simulation montrent bien la poursuite de trajectoire pendant le parcourt des trois modes de fonctionnement avec une amélioration considérable du temps de calcul de la solution optimale du problème quadratique linéaire mixte entier. L'utilisation de cette configuration permet une implémentation en temps réel avec une période d'échantillonnage petite sans recourt un solveur puissant.



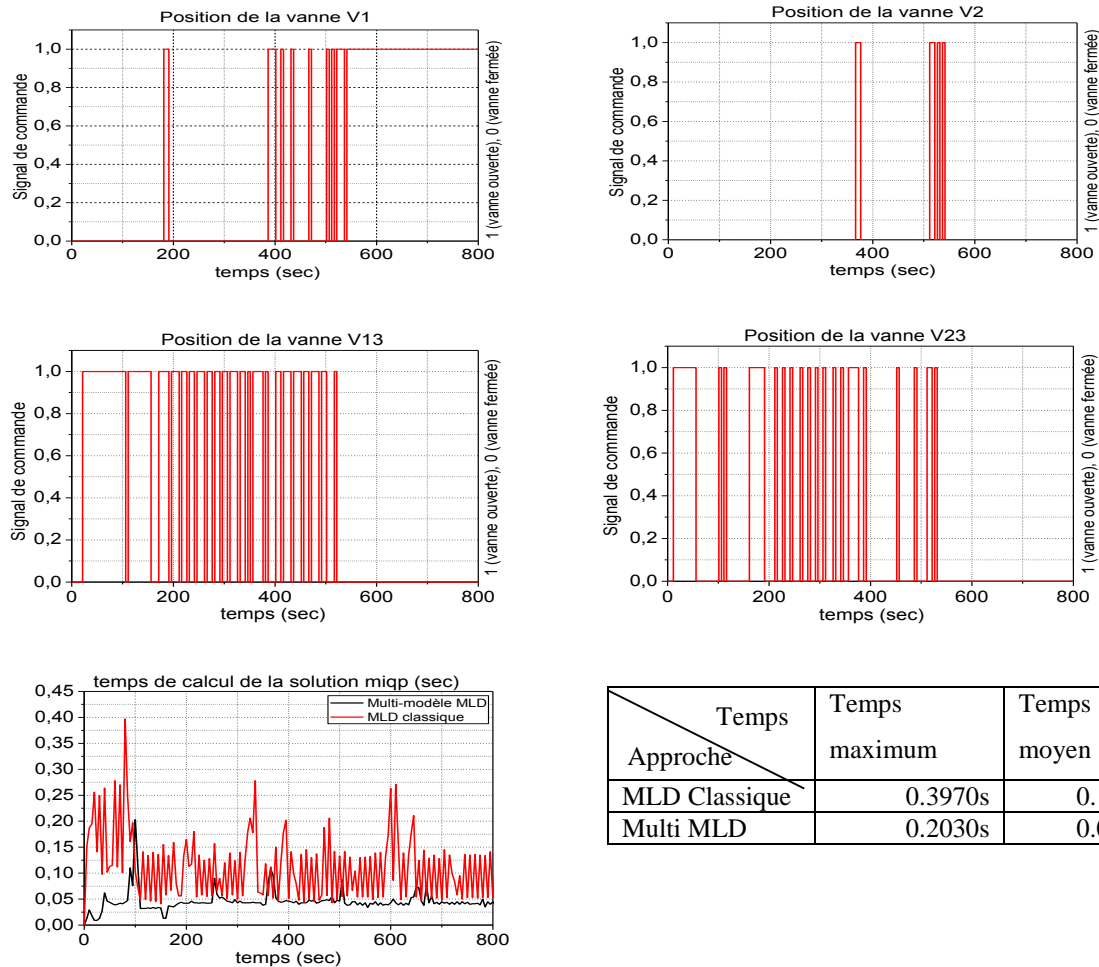


Figure IV.11 : Commande prédictive multi-modèles des niveaux  $h_1$ ,  $h_2$  et  $h_3$

(Horizon de prédiction  $N=3$ ,  $Q_u=610^3$ ,  $Q_d=Q_z=10^{-2}$ , vannes  $V_{L1}$ ,  $V_{L2}$  et  $V_{L3}$  sont contrôlées avec variation en échelon des consignes des niveaux  $h_1$ ,  $h_2$  et  $h_3$ )

Enfin, pour terminer avec la commande prédictive hybride **MPC**, on donne les résultats obtenus par simulation de l'application sur un benchmark sur un modèle **APROS** à trois réservoirs de la configuration d'une commande MPC avec Multi-modèle MLD (voir la figure IV.10). La simulation effectuée contient trois modes opératoires **M1**, **M2** et **M3**, chacun de ces modes appartient un espace limité par les contraintes physique du benchmark. Le mode **M1** est choisi dans l'espace de fonctionnement  $[0-0.25] \times [0-0.25] \times [0-0.1](m^3)$ , le mode **M2** est choisi dans l'espace  $[0.25-0.45] \times [0.25-0.45] \times [0-0.20](m^3)$  et le mode **M3** est choisi dans l'espace de travail  $[0.45-0.55] \times [0.45-0.55] \times [0.20-0.30](m^3)$  avec une configuration initiale. Les résultats de simulation montrent bien la poursuite de trajectoire pendant le parcourt des trois modes de fonctionnement avec une amélioration considérable du temps de calcul de la solution optimale du problème quadratique linéaire mixte entier. L'utilisation de cette configuration permet une implémentation en temps réel avec une période d'échantillonnage petite sans recourt un solveur puissant.

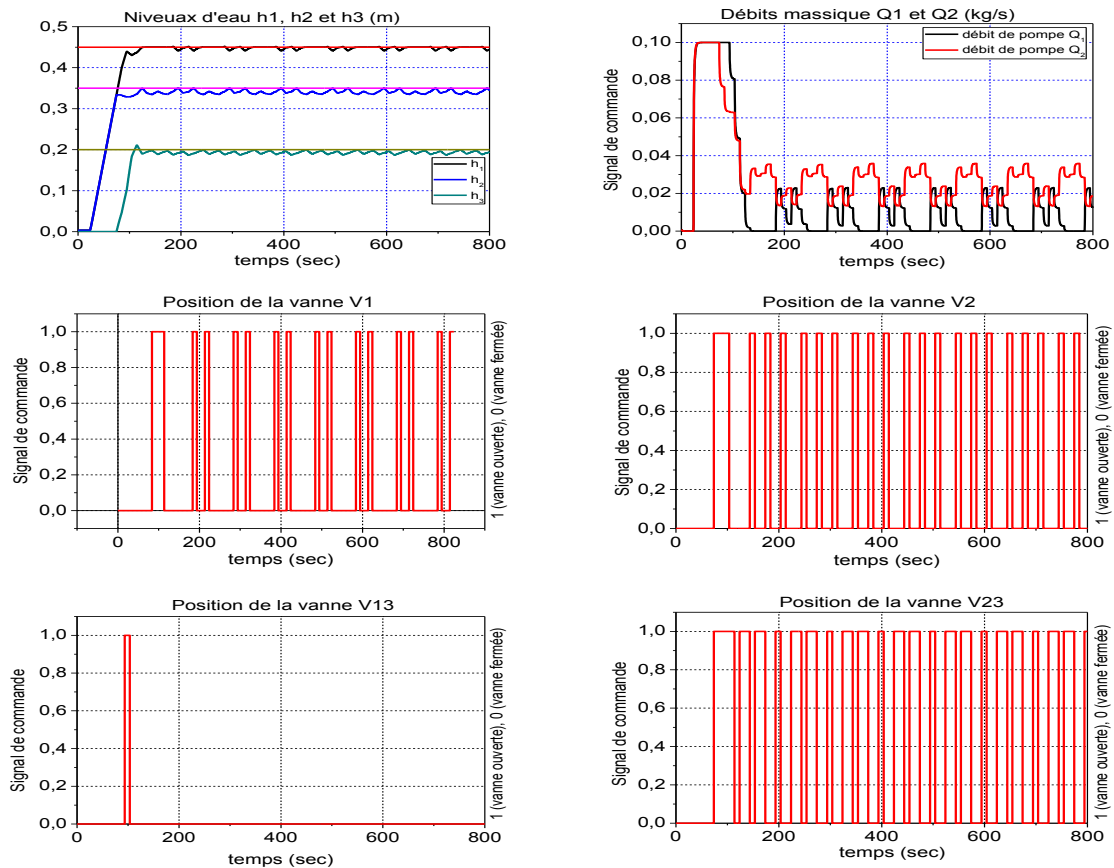


Figure IV.12 : Commande prédictive affine par morceaux PWA des niveaux  $h_1$ ,  $h_2$  et  $h_3$   
 (Horizon de prédiction  $N=3$ ,  $Q_y=10^3$ ,  $Q_u=10^{-3}$ , vannes  $V_{L3}$  est maintenue ouverte avec variation en échelon des consignes des niveaux  $h_1$ ,  $h_2$  et  $h_3$ )

Enfin, la dernière configuration concerne l'approche hybride explicite avec l'utilisation de la boîte à outil MPT multiparamétrique (voir figure IV.12). Le modèle affine par morceaux équivalent est obtenu à l'aide de la boîte multiparamétrique MPT par une transformation du modèle MLD vers le modèle PWA par un partitionnement de l'espace d'état en régions d'espace polyhédral associé à une équation d'état affine. Pour ce type de classe de système, le problème de commande sous contraintes est résolu hors-ligne en utilisant la boîte d'outil MPT (MultiParametric Toolbox). La solution du problème de commande est une loi par retour d'état affine où les gains sont calculés par une optimisation  $mp\_MILP$ . Le calcul de la commande revient à déterminer l'appartenance à la région qui permet de déterminer les gains associés à cette dernière par l'intermédiaire d'une table lookup à l'évaluation du retour d'état à chaque période d'échantillonnage.

La boîte à outil multi-Parametric Toolbox, nous a permis de calculer le contrôleur PWA défini sur 1216 régions. Le nombre de régions dépend sur la l'horizon de prédiction horizon, les contraintes fonctionnels et le nombre des deux dynamiques et les variables d'état du modèle affine PWA. Les résultats obtenus sont très satisfaisantes en poursuite de trajectoire.

**V.5 Conclusion :**

Nous avons présenté dans ce chapitre les résultats de simulation suite à l'application de trois approches de commande au modèle trois cuves sous l'environnement APROS. La commande MPC utilisant un modèle MLD pour la prédiction du comportement dynamique du système hybride avec un algorithme d'optimisation MIQ assure une poursuite de trajectoire seulement avec une charge de calcul augmentée, qui implique le recours à des solveurs performant ou faire un découpage de l'espace d'état en plusieurs régions. Une autre alternatif pour remédier à ce problème est d'utilisé une méthode d'optimisation hors-ligne. Les résultats obtenus sont commentés pour différentes scénarios de simulation.

# Conclusion Générale

# Conclusion générale & Perspectives

Le présent travail dans le cadre du mémoire de magister consiste en la modélisation, à l'aide du logiciel APROS, d'un système non-linéaire composé de trois réservoirs interconnectés en vue d'une commande hybride par deux approches prédictive à base de modèle (MPC) en poursuite de référence. Les approches de commande s'appuient sur le formalisme MLD, qui prend en compte des dynamiques, des états, des entrées, des sorties et des contraintes continues et binaires ainsi que des interfaces entre ces deux aspects. Il permet de modéliser plusieurs classes de systèmes hybrides, telles que les systèmes hybrides linéaires, les systèmes linéaires sous contraintes, les systèmes dynamiques non-linéaires où les non linéarités sont exprimés à travers des combinaisons logiques. Les stratégies de commande prédictive à horizon glissant (MPC) développées sont essentiellement basées sur des techniques d'optimisation pour lesquelles on cherche à minimiser une fonction de coût quadratique ou linéaire, résolvant des problèmes de programmation mixte (MIQP, MILP).

En premier, nous nous sommes intéressée à l'étude des concepts et modèles des systèmes dynamiques hybrides, ce qui nous a permis d'aborder notre problème de modélisation, simulation et commande. La seconde partie du travail a été consacrée à l'utilisation du code système APROS pour la modélisation du système choisi afin de décrire une dynamique hybride inclus dans le fonctionnement du benchmark COSY à trois cuves. L'utilisation de l'environnement **APROS** nécessite une connaissance approfondie du principe de la **nodalisation** ou du découpage, afin d'établir un modèle du procédé. Ce dernier peut contenir des simples composants décrivant un mouvement de fluide à l'état liquide avec des pompes, des vannes, des coudes, des régulateurs de débit, de niveau, de température...etc., jusqu'au modèle le plus complexe dans une installation. La maîtrise de l'outil APROS nous a permis de mettre en place le modèle de notre application benchmark à trois cuves **COSY**.

La commande prédictive à modèle de référence avec un horizon fuyant, a été abordée par simulation en utilisant l'outil de modélisation **HYSDEL**. La synthèse de la commande **MPC** à horizon glissant nécessite dans un premier temps de faire une translation du modèle physique régi par les équations dynamique du fluide et par les contraintes fonctionnelles. Le système fait intervenir des composants avec une dynamique continue et d'autres avec une dynamique discrète. Cette complexité du système par la présence de deux dynamiques fait apparaître la nature d'une dynamique hybride, qui ne pourra être résolu que par l'utilisation d'un formalisme permettant d'avoir un modèle hybride **MLD**. Une fois le modèle est établi, la synthèse de la commande prédictive à horizon glissant est reformulée sous forme d'un problème d'optimisation quadratique linéaire avec contraintes linéaires en présence de variables réelles et entières MIQP. La résolution du problème a été surmontée par l'utilisation du solveur **CPLEX**.

La seconde approche **MPC** avec une optimisation offline a été dédiée à la synthèse d'une commande optimale explicite en utilisant la bibliothèque MultiParametric Toolbox **MPT**.

L'application de cette commande après la phase d'optimisation par l'utilisation d'une table **lookup** qui sert au stockage du régulateur piecewise PWA dans les différentes régions polyèdre.

Les trois approches commandes MPC sous MLD, multi-MLD et PWA ont été appliquées sur le modèle **APROS** à trois cuves par l'interface **MATLAB-OPC**. Pour cela, nous avons réalisé une interface logicielle permettant l'échange de données et l'envoi de commande.

Les résultats de simulations confirment la poursuite parfaite de la trajectoire de référence pour le contrôle des niveaux de liquide des trois cuves.

Enfin, le travail effectué dans le cadre de ce mémoire ouvre de nombreuses voies de recherches telles que :

- La commande non-linéaire hybride avec une optimisation non-linéaire,
- La commande linéaire et non-linéaire hybride par les algorithmes génétiques,
- Identification des systèmes dynamiques hybrides,
- Etude des systèmes dynamiques hybrides dans un milieu bruité,

# Annexes



# Annexe A

Tableau A.1 : Les outils en-ligne de modélisation, simulation et optimisation pour les systèmes dynamiques hybrides [39].

Produit	Fournisseur	Licence	Site web
20-sim	Controllab Products B.V, Netherlands	Commercial	ww.2sim.com
ABACUSS II	MIT, USA	Free	http://yoric.mit.edu
AMPL	AMPL Optimization LLC, USA	Commercial	www.ampl.com
BaSip	TU Dortmund, Germany	Free	www.bci.tu-dortmund.de/basip
Charon	University of Pennsylvania, USA	Free	http://rtg.cis.upenn.edu/mobies/charon
CheckMate	Carnegie Mellon University, USA	Free	www.ece.cmu.edu/webk/checkmate
$\chi$	TU Eindhoven, Netherlands	Free	http://se.wtb.tue.nl/sewiki/chi
Dymola	Dynasim AB, Sweden	Commercial	www.dynasim.se
EcosimPro	EA Internacional, Spain	Commercial	www.ecosimpro.com
GAMS	GAMS Software GmbH, Germany	Commercial	www.gams.com
gPROMS/gOPT	Process Systems Enterprise Ltd, UK	Commercial	www.psenderprise.com/gproms
HQP	TU Ilmeau, Germany	Free	http://sourceforge.net/projects/hqp
IPOPT	Carnegie Technology, LLC, USA	Free	www.coin-or.org/Ipopt
Jacobian	Numerical Technology, LLC, USA	Commercial	www.numericatech.com
MathModelica	MathCore Engineering AB, Sweden	Commercial	www.mathcore.com/products
MATLAB/Simulink/ Stateflow/SimEvents	The MathWorks, USA	Commercial	www.mathworks.com
MUSCOD II	Universitdt Heidelberg, Germany	Commercial	www.iwr.uni-heidelberg.de/~agbock/RESEARCH/muscod.php
Omola/Omsim	Lunds Universitet, Sweden	Free	www.control.lth.se/~cace/omsim.html
Optimica	Lunds Universitet, Sweden	Free	www.control.lth.se/project/Langopt
OpenModelica	Linkvings Universitet, Sweden	Free	www.openmodelica.org
Ptolemy II/HyVisual	UC Berkeley, USA	Free	http://ptolemy.eecs.berkeley.edu
Scilab/Scicos	INRIA, France	Free	www.scilab.org, www.scicos.org
Shift	UC Berkeley, USA	Free	www.path.berkeley.edu/shift
Siconos	INRIA, France	Free	http://siconos.gforge.inria.fr
TOMLAB	TOMLAB Optimization	Commercial	www.tomopt.com
UPPAAL	Uppsala/Aalborg Univ., Sweden/Denmark	Free	www.uppaal.com
RELAP	Innovative Systems Software, USA	Commercial	http://www.relap.com/
CATHARE	CEA, France	Commercial	http://www-cathare.cea.fr/
APROS	VTT, Finlande	Commercial	http://www.apros.fi

# Annexe B

## Le langage de modélisation HYSDEL

### B.1 Objectif et principe d'utilisation :

La transformation de la description initiale d'un système hybride en une forme **MLD** exige l'application d'un ensemble de règles données. Il est maintenant nécessaire d'adopter un outil qui permet d'obtenir cette forme **MLD** de façon systématique, évitant ainsi des calculs fastidieux. Pour cela, un compilateur est développé, fournissant après calcul les matrices  $A, B_1, B_2, B_3, C, D_1, D_2, D_3, E_1, E_2, E_3, E_4, E_5$  de l'équation (III.6). Le langage spécifiant le problème au compilateur **HYSDEL** (Hybrid System Description Langage)[1].

L'appel au compilateur se fit sous MATLAB, par l'intermédiaire d'un fichier source « \*.hys ». Il résulte de la compilation un fichier « \*.m » dans lequel la forme **MLD** complète du système. Ce fichier regroupe donc la formalisation d'un « Object MATLAB » toutes les matrices de la forme **MLD** comme suit :

$$\begin{cases} \begin{bmatrix} x_c(k+1) \\ x_l(k+1) \end{bmatrix} = \begin{bmatrix} A_{cc} & A_{cd} \\ A_{dc} & A_{dd} \end{bmatrix} \begin{bmatrix} x_c(k) \\ x_l(k) \end{bmatrix} + \begin{bmatrix} B_{1cc} & B_{1cd} \\ B_{1dc} & B_{1dd} \end{bmatrix} \begin{bmatrix} u_c(k) \\ u_l(k) \end{bmatrix} + \begin{bmatrix} B_{2cd} \\ B_{2dd} \end{bmatrix} \delta(k) + \begin{bmatrix} B_{3cd} \\ B_{3dd} \end{bmatrix} z(k) \\ \begin{bmatrix} y_c(k+1) \\ y_l(k+1) \end{bmatrix} = \begin{bmatrix} C_{cc} & C_{cd} \\ C_{dc} & C_{dd} \end{bmatrix} \begin{bmatrix} x_c(k) \\ x_l(k) \end{bmatrix} + \begin{bmatrix} D_{1cc} & D_{1cd} \\ D_{1dc} & D_{1dd} \end{bmatrix} \begin{bmatrix} u_c(k) \\ u_l(k) \end{bmatrix} + \begin{bmatrix} D_{2cd} \\ D_{2dd} \end{bmatrix} \delta(k) + \begin{bmatrix} D_{3cd} \\ D_{3dd} \end{bmatrix} z(k) \\ E_2 \delta(k) + E_3 z(k) \leq E_1 u(k) + E_4 x(k) + E_5 \end{cases} \quad (\text{B.1})$$

Le fichier final inclut également un certain nombre de variables caractérisant les dimensions des matrices.

### B.2 Description du langage HYSDEL :

**HYSDEL** est un langage permettant de traiter un système hybride selon la structure **MLD** à partir d'une description textuelle de ce système. Chaque fichier est en réalité composé de deux parties. La première partie, appelée « **Interface** » a pour but de déclarer toutes les variables : Les états, les entrées, les sorties, ainsi que toutes les constantes. La deuxième partie, dite « **mise en œuvre** », est composée de six sections spécialisées décrivant les relations existant entre les variables, de la façon suivante :

1. **Section «AD»** : elle définit les variables auxiliaires logiques issues de l'interprétation de relations continues. On trouve ainsi dans cette section des relations :  $\delta = 1$  si  $a_1 x_1 + a_2 x_2 \leq b$ .
2. **Section «DA»** : Elle définit les variables auxiliaires continues, issues de la valeur prise par des variables booléennes. On trouve alors des relations: si  $\delta = 1$  alors  $z = a_1 x + b_1$  sinon  $z = a_2 x + b_2$ .
3. **Section «LOGIC»** : elle définit les fonctions impliquant des variables booléennes. Ainsi on pourra y inclure des relations du type :  $\{\delta = \delta_1 \& (\delta_2 \mid \delta_3)\}$ .

4. **Section « CONTINUOUS »** : elle décrit la dynamique linéaire, sous la forme d'équation aux différences, par exemple :  $h(k+1) = h(k) + \frac{T_s}{A}(Q_1 - Q_2)$ .
5. **Section « AUTOMATA »** : Cette section spécifie les équations caractérisant les transitions des états de l'automate. Ainsi, on trouvera des relations :  $V = (X_1 \& a) | (X_2 \& b)$ .
6. **Section « MUST »** : Cette section spécifie les contraintes sur les variables continues et binaires (contraintes linéaires et formules booléennes), par exemple:  $x_{\min} \leq x \leq x_{\max}, \{(u_1 \& \sim u_2) | (\sim u_1 \& u_2)\}$

Enfin le modèle MLD obtenu n'est pas unique, dépendant de la façon d'introduire les variables auxiliaires. Il existe en effet de multiples formes MLD pour un même système, il est alors préférable de rechercher celui qui procure le nombre de variables logiques le plus faible. Néanmoins, cet outil permet de fournir automatiquement des modèles MLD s'avère très utile pour la définition de la forme d'état associée.

# Annexe C

Programme HYSDEL du modèle MLD classique pour le benchmark COSY « trois cuves »

```
SYSTEM Tanks {
INTERFACE {
STATE {
    REAL h1,h2,h3;
}
INPUT {
    REAL Q1,Q2;
    BOOL V1,V2,V13,V23;
}
PARAMETER
{
REAL Su = 0.0154;
REAL Ts = 5;
REAL k1 = 0.0001125; /* constant for valves V13, V23, VL1 and VN3 */
REAL k2 = 0.0001566; /* constant for valves V1 and V2 */
REAL hmax = 0.62;
REAL hv = 0.30;
REAL Qmax = 0.0001;
REAL VL1 = 0;
REAL VN3 = 1;
}
IMPLEMENTATION {
AUX {
    REAL z01,z02,z03,z1,z2,z13,z23;
    BOOL d01,d02,d03;
}
AD
{
    d01 = hv-h1 <= 0;
    d02 = hv-h2 <= 0;
    d03 = hv-h3 <= 0;
}
DA
{
    z01={ IF d01 THEN h1-hv ELSE 0};
    z02={ IF d02 THEN h2-hv ELSE 0};
    z03={ IF d03 THEN h3-hv ELSE 0};
    z1={ IF V1 THEN z01-z03 ELSE 0};
```

```

z2={ IF V2 THEN z02-z03 ELSE 0};
z13={ IF V13 THEN h1-h3 ELSE 0};
z23={ IF V23 THEN h2-h3 ELSE 0};
}

CONTINUOUS
{
h1=h1+(Ts/Su)*(Q1-k1*z13-k2*z1-k1*h1*VL1);
h2=h2+(Ts/Su)*(Q2-k1*z23-k2*z2);
h3=h3+(Ts/Su)*(k1*z13+k1*z23+k2*z1+k2*z2-k1*h3*VN3);
}

MUST
{
h1<=hmax;
h2<=hmax;
h3<=hmax;
-h1<=0;
-h2<=0;
-h3<=0;
Q1<=Qmax;
Q2<=Qmax;
-Q1<=0;
-Q2<=0;
}
}
}

```

# Annexe D

Annexe B-1 :

Programme HYSDEL du modèle MLD **multiple** pour le benchmark COSY « trois cuves » pour le premier domaine **M1**.

```
SYSTEM Tanks {  
INTERFACE {  
STATE {  
    REAL h1,h2,h3;  
}  
INPUT {  
    REAL Q1,Q2;  
    BOOL V1,V2,V13,V23;  
}  
PARAMETER  
{  
REAL Su = 0.0154;  
REAL Ts = 5;  
REAL k1 = 0.0001125; /* constant for valves V13, V23, VL1 and VN3 */  
REAL k2 = 0.0001566; /* constant for valves V1 and V2 */  
REAL hmax = 0.62;  
REAL hv = 0.30;  
REAL Qmax = 0.0001;  
REAL VL1 = 0;  
REAL VN3 = 1;  
}  
}  
IMPLEMENTATION {  
AUX {  
    REAL z13,z23;  
}  
DA  
{  
    z13={ IF V13 THEN h1-h3 ELSE 0};  
    z23={ IF V23 THEN h2-h3 ELSE 0};  
}  
CONTINUOUS  
{  
    h1=h1+(Ts/Su)*(Q1-k1*z13-k2*z1-k1*h1*VL1);  
    h2=h2+(Ts/Su)*(Q2-k1*z23-k2*z2-k2*h2*VL2);  
    h3=h3+(Ts/Su)*(k1*z13+k1*z23+k2*z1+k2*z2-k1*h3*VN3);
```

}

MUST

{

$h1 \leq hmax;$

$h2 \leq hmax;$

$h3 \leq hmax;$

$-h1 \leq 0;$

$-h2 \leq 0;$

$-h3 \leq 0;$

$Q1 \leq Qmax;$

$Q2 \leq Qmax;$

$-Q1 \leq 0;$

$-Q2 \leq 0;$

}

}

}

Annexe B-2:

Programme HYSDEL du modèle MLD **multiple** pour le benchmark COSY « trois cuves » pour le deuxième domaine **M2**.

```
SYSTEM Tanks {  
  
INTERFACE {  
  
STATE {  
    REAL h1,h2,h3;  
}  
  
INPUT {  
    REAL Q1,Q2;  
    BOOL V1,V2,V13,V23;  
}  
  
PARAMETER  
{  
REAL Su = 0.0154;  
REAL Ts = 5;  
REAL k1 = 0.0001125; /* constant for valves V13, V23, VL1 and VN3 */  
REAL k2 = 0.0001566; /* constant for valves V1 and V2 */  
REAL hmax = 0.62;  
REAL hv = 0.30;  
REAL Qmax = 0.0001;  
REAL VL1 = 0;  
REAL VN3 = 1;  
}  
}  
IMPLEMENTATION {  
  
AUX {  
    REAL z1,z2,z13,z23;  
    BOOL d01,d02;  
}  
  
AD  
{  
    d01 = hv-h1 <= 0;  
    d02 = hv-h2 <= 0;  
    d03 = hv-h3 <= 0;  
}  
  
DA  
{  
    z1={ IF V1 THEN h1-hv ELSE 0};  
    z2={ IF V2 THEN h2-hv ELSE 0};  
    z13={ IF V13 THEN h1-h3 ELSE 0};  
    z23={ IF V23 THEN h2-h3 ELSE 0};  
}  
}
```



CONTINUOUS

```
{  
  h1=h1+(Ts/Su)*(Q1-k1*z13-k2*z1-k1*h1*VL1);  
  h2=h2+(Ts/Su)*(Q2-k1*z23-k2*z2-k2*h2*VL2);  
  h3=h3+(Ts/Su)*(k1*z13+k1*z23+k2*z1+k2*z2-k1*h3*VN3);  
}
```

MUST

```
{  
  h1<=hmax;  
  h2<=hmax;  
  h3<=hmax;  
  -h1<=0;  
  -h2<=0;  
  -h3<=0;  
  Q1<=Qmax;  
  Q2<=Qmax;  
  -Q1<=0;  
  -Q2<=0;  
}  
}  
}
```

# Annexe E

Formulation de l'algorithme d'optimisation quadratique MPC à horizon glissant sous forme d'un problème **MIQP** :

Le modèle MLD décrivant le système trois-cuves est donné par :

$$\begin{cases} x(t+1) = Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) \\ y(t) = Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) \\ E_2\delta(t) + E_3z(t) \leq E_1u(t) + E_4x(t) + E_5 \end{cases} \quad (\text{C.1})$$

A partir du modèle MLD, la commande prédictive à base de modèle est considérée de la manière suivante :

$$\begin{aligned} \min_{\{u_0^{N-1}\}} J(u_0^{N-1}, x_0) = & \sum_{k=0}^{N-1} \|u(k) - u_e\|_{Q_1}^2 + \|\delta(k/t) - \delta_e\|_{Q_2}^2 + \|z(k/t) - z_e\|_{Q_3}^2 \\ & + \|x(k+1/t) - x_e\|_{Q_4}^2 + \|y(k/t) - y_e\|_{Q_5}^2 \end{aligned} \quad (\text{C.2})$$

Est soumis à l'équation (C.1).

Le critère d'optimisation précédent peut être réécrit comme suit :

$$\min_{\{u_0^{N-1}\}} J = \|T - T_e\|_Q = (T - T_e)^T Q (T - T_e) \quad (\text{C.3})$$

Où :

$$Q^T = \text{diag}[\text{diag}(Q_4)_N \text{diag}(Q_5)_N \text{diag}(Q_1)_N \text{diag}(Q_2)_N \text{diag}(Q_3)_N]$$

Où  $\text{diag}(Q_i)_N$  est une matrice diagonale d'éléments  $Q_i$  et de dimension  $N$ .

$$\begin{aligned} T = & \left[ x(k+1)^T, \dots, x(k+N)^T, y(k)^T, \dots, y(k+N-1)^T, u(k)^T, \dots, u(k+N-1)^T, \right. \\ & \left. \delta(k)^T, \dots, \delta(k+N-1)^T, z(k)^T, \dots, z(k+N-1)^T \right] \end{aligned} \quad (\text{C.4})$$

Et que  $T_e$  représente le vecteur des points d'équilibre.

On utilisant la forme MLD (A.1), les variables  $(x, y)$  sont remplacées par leurs équivalente en  $(u, \delta, z)$

. Par conséquent, la forme MLD devient :

$$T = P\chi + \varphi \quad (\text{C.5})$$

$$\begin{bmatrix} x_1 \\ \vdots \\ x_N \\ y_0 \\ \vdots \\ y_{N-1} \\ u_0 \\ \vdots \\ u_{N-1} \\ \delta_0 \\ \vdots \\ \delta_{N-1} \\ z_0 \\ \vdots \\ z_{N-1} \end{bmatrix} \underset{T}{=} \begin{bmatrix} B_1 & \cdots & 0 & B_2 & \cdots & 0 & B_3 & \cdots & 0 \\ AB_1 & \ddots & & \ddots & & & & \ddots & \\ A^{N-1}B_1 & \cdots & B_1 & A^{N-1}B_2 & \cdots & B_2 & A^{N-1}B_3 & \cdots & B_3 \\ D_1 & \cdots & 0 & D_2 & \cdots & 0 & D_3 & \cdots & 0 \\ CB_1 & & & CB_2 & & & CB_3 & & \\ CA^{N-1}B_1 & \cdots & D_1 & CA^{N-1}B_2 & \cdots & D_2 & CA^{N-1}B_3 & \cdots & D_3 \\ I_{m \times m} & & & & & & & & \\ & I_{m \times m} & & & & & & & \\ & & I_{m \times m} & & & & & & \\ & & & I_{r_1 \times r_1} & & & & & \\ & & & & I_{r_1 \times r_1} & & & & \\ & & & & & I_{r_1 \times r_1} & & & \\ & & & & & & I_{r_c \times r_c} & & \\ & & & & & & & I_{r_c \times r_c} & \\ & & & & & & & & I_{r_c \times r_c} \end{bmatrix} \begin{bmatrix} u_0 \\ \vdots \\ u_{N-1} \\ \delta_0 \\ \vdots \\ \delta_{N-1} \\ z_0 \\ \vdots \\ z_{N-1} \end{bmatrix} \underset{\chi}{=} \begin{bmatrix} Ax_0 \\ A^2x_0 \\ \vdots \\ A^Nx_0 \\ Cx_0 \\ CAx_0 \\ \vdots \\ CA^Nx_0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \underset{\varphi}{=}$$

De l'équation (C.5) et (C.3)

$$\begin{aligned} \min_{\{u_0^{N-1}\}} J &= \left\| \begin{matrix} P\chi + \varphi - T_e \\ \chi_e \end{matrix} \right\|_Q = (P\chi - \chi_e)^T Q (P\chi - \chi_e) \\ &= \chi^T P^T Q P \chi - 2\chi_e^T Q P \chi + \chi_e^T Q \chi_e \end{aligned} \quad (C.6)$$

Le troisième terme est constant, ainsi l'équation (A.6) peut être écrite comme suite :

$$\min_{\{u_0^{N-1}\}} J = \frac{1}{2} \chi^T H \chi + f^T \chi \quad (C.7)$$

Où :  $H = P^T Q P$  et  $f^T = \chi_e^T Q P$

Le problème d'optimisation (A.7) est résolu sous la contrainte :  $c\chi \leq b$

Où : c, b sont calculées par l'équation d'inégalités du modèle MLD (A.1), on remplaçant la variable  $x$  par  $(u, \delta, z)$  dans les contraintes d'égalités (A1).

$$\begin{cases} E_2\delta(k) + E_3z(k) \leq E_1u(k) + E_4x(k) + E_5 \\ E_2\delta(k+1) + E_3z(k+1) \leq E_1u(k+1) + E_4x(k+1) + E_5 \\ \vdots \\ E_2\delta(k+N) + E_3z(k+N) \leq E_1u(k+N) + E_4x(k+N) + E_5 \end{cases} \quad (C.8)$$

Pour faire ressortir le vecteur  $\chi$ , on procède à un réarrangement sur les équations de contraintes en remplaçant la valeur de la variable  $x_{(k+j)}$  par son équivalent :

$$\begin{cases} -E_1 u(k) + E_2 \delta(k) + E_3 z(k) \leq E_4 x(k) + E_5 \\ -E_1 u(k+1) + E_2 \delta(k+1) + E_3 z(k+1) \leq E_4 (Ax(k) + B_1 u(k) + B_2 \delta(k) + B_3 z(k)) + E_5 \\ \vdots \\ -E_1 u(k+N) + E_2 \delta(k+N) + E_3 z(k+N) \leq E_4 x(k+N) + E_5 \end{cases} \quad (C.9)$$

Après simplification, on obtient la forme suivante :

$$\begin{cases} -E_1 u(k) + E_2 \delta(k) + E_3 z(k) \leq E_4 x(k) + E_5 \\ -E_4 B_1 u(k) - E_1 u(k+1) - E_4 B_2 \delta(k) + E_2 \delta(k+1) - E_4 B_3 z(k) + E_3 z(k+1) \leq E_4 Ax(k) + E_5 \\ \vdots \\ -E_4 A^{N-1} B_1 u(k) - \dots - E_1 u(k+N-1) - E_4 A^{N-1} B_1 \delta(k) + \dots + E_2 \delta(k+N-1) \\ -E_4 A^{N-1} B_1 z(k) + \dots + E_3 z(k+N-1) \leq E_4 x(k+N) + E_5 \end{cases} \quad (C.10)$$

A partir de cette forme, on aura l'écriture compacte matricielle de l'expression de la contrainte du problème MIQP  $c\chi \leq b$  :

$$\chi = [u(k)^T, \dots, u(k+N-1)^T, \delta(k)^T, \dots, \delta(k+N-1)^T, z(k)^T, \dots, z(k+N-1)^T]^T$$

$$c = \begin{bmatrix} -E_1 & \dots & 0 & E_2 & \dots & 0 & E_3 & \dots & 0 \\ -E_4 B_1 & & & -E_4 B_2 & & & -E_4 B_3 & & \\ -E_4 A B_1 & \ddots & & -E_4 A B_2 & \ddots & & -E_4 A B_3 & \ddots & \\ -E_4 A^2 B_1 & & & -E_4 A^2 B_2 & & & -E_4 A^2 B_3 & & \\ \vdots & & & \vdots & & & \vdots & & \vdots \\ -E_4 A^{N-2} B_1 & \dots & -E_1 & -E_4 A^{N-2} B_2 & \dots & E_2 & -E_4 A^{N-2} B_3 & & E_3 \end{bmatrix}$$

$$b^T = [(E_4 x(k) + E_5)^T \quad (E_4 Ax(k) + E_5)^T \quad \dots \quad (E_4 A^{N-1} x(k) + E_5)^T]$$

# Annexe F

Formulation de l'algorithme d'optimisation quadratique MPC à horizon glissant sous forme d'un problème **MIQP** présenté par Bemporad et Morari [6] :

Le modèle MLD décrivant le système trois-cuves est donné par :

$$\begin{cases} x_{k+1} = Ax_k + B_1 u_k + B_2 \delta_k + B_3 z_k \\ y_k = Cx(k) \end{cases} \quad (\text{E.1})$$

$$\text{subject to : } E_2 \delta_k + E_3 z_k \leq E_1 u_k + E_4 x_k + E_5 \quad (\text{E.2})$$

A partir du modèle MLD, la commande prédictive à base de modèle est considérée de la manière suivante :

$$\begin{aligned} \min_{u, \delta, z} J(\left\{ \begin{array}{l} u_k \cdots u_{k+m-1} \\ \delta_k \cdots \delta_{k+p-1} \\ z_k \cdots z_{k+p-1} \end{array} \right\}, x(k)) = & \sum_{i=1}^{N_p} \left( \|y_{k+i} - y_{sp}\|_{Q_5}^2 + \|\delta_{k+i} - \delta_{sp}\|_{Q_2}^2 + \|z_{k+i} - z_{sp}\|_{Q_3}^2 \right) \\ & + \sum_{i=0}^{N_u-1} \left( \|u_{k+i} - u_{k+i-1}\|_{Q_u}^2 + \|u_{k+i} - u_{sp}\|_{Q_1}^2 \right) \end{aligned} \quad (\text{E.3})$$

Est soumis à la contrainte définie par l'équation (E.2).

Les prescripteurs *sp*, *min* et *max* définissent la référence, la borne inférieure et la borne supérieure pour chaque vecteur. Dans le but de simplifier la fonction du critère d'optimisation, les vecteurs de prédiction et de contrôle peuvent être définis comme suit :

$$\psi_k = [y_k \ y_{k+1} \cdots y_{k+p}]^T \quad (\text{E.4})$$

$$\mu_k = [\mu_k \ \mu_{k+1} \cdots \mu_{k+m-1}]^T \quad (\text{E.5})$$

$$\bar{\delta}_k = [\delta_k \ \delta_{k+1} \cdots \delta_{k+p-1}]^T \quad (\text{E.6})$$

$$\bar{z}_k = [z_k \ z_{k+1} \cdots z_{k+p-1}]^T \quad (\text{E.7})$$

$$\chi = [\mu_k \ \bar{\delta}_k \ \bar{z}_k]^T \quad (\text{E.8})$$

$\zeta^k$  est la perturbation à l'instant  $k$ .

En utilisant les notations approuvées sur les vecteurs de commande et prédiction, le critère (E.3) peut se réécrire de la façon suivante :

$$\begin{aligned}
\min_{u, \delta, z} J \left( \begin{array}{l} u_k \cdots u_{k+m-1} \\ \delta_k \cdots \delta_{k+p-1} \\ z_k \cdots z_{k+p-1} \end{array} \right), x(k) &= (\psi_k - \psi_{sp})^T Q_5 (\psi_k - \psi_{sp}) + (\bar{\delta}_k - \bar{\delta}_{sp})^T Q_2 (\bar{\delta}_k - \bar{\delta}_{sp}) \\
&+ (\bar{z}_k - \bar{z}_{sp})^T Q_2 (\bar{z}_k - \bar{z}_{sp}) + (\mu_k - \mu_{k-1})^T Q_4 (\mu_k - \mu_{k-1}) \\
&+ (\mu_k - \mu_{sp})^T Q_1 (\mu_k - \mu_{sp})
\end{aligned} \tag{E.9}$$

La valeur du vecteur de sortie  $\psi_k$  est remplacée par l'équation d'état décrivant le système hybride (E.1) avec la forme prédictive suivante :

$$\psi_k = H_{M1}\mu_k + H_{M2}\bar{\delta}_k + H_{M3}\bar{z}_k + \Phi_M \bar{x}_k \tag{E.10}$$

Avec :

$$\Phi_M = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^p \end{bmatrix} \tag{E.11}$$

$$H_{M1} = \begin{bmatrix} CB_1 & 0 & \dots & & 0 \\ CAB_1 & CB_1 & \dots & & 0 \\ CA^2 B_1 & CAB_1 & \dots & & 0 \\ \vdots & \vdots & & & \vdots \\ CA^{(m-1)} B_1 & CA^{(m-2)} B_1 & \dots & CAB_1 & CB_1 \\ CA^m B_1 & CA^{(m-1)} B_1 & \dots & CA^2 B_1 & CB_1 + CAB_1 \\ \vdots & \vdots & & \vdots & \vdots \\ CA^{(p-1)} B_1 & CA^{(p-2)} B_1 & \dots & CA^{(p-m-1)} & \sum_{i=0}^{p-m} CA^i B_1 \end{bmatrix} \tag{E.12}$$

et

$$H_{Mi} = \begin{bmatrix} CB_i & 0 & \dots & & 0 \\ CAB_i & CB_i & \dots & & 0 \\ CA^2 B_i & CAB_i & \dots & & 0 \\ \vdots & \vdots & & & \vdots \\ CA^{(p-2)} B_i & CA^{(p-3)} B_i & \dots & CB_i & 0 \\ CA^{(p-1)} B_i & CA^{(p-2)} B_i & \dots & CAB_i & CB_i \end{bmatrix} \text{ where } i=2,3 \tag{E.13}$$

$$\begin{aligned}
\min_{u, \delta, z} J &= (H_{M1}\mu_k + H_{M2}\bar{\delta}_k + H_{M3}\bar{z}_k + \Phi_M \bar{x}_k - \psi_{sp})^T Q_5 (H_{M1}\mu_k + H_{M2}\bar{\delta}_k + H_{M3}\bar{z}_k + \Phi_M \bar{x}_k - \psi_{sp}) \\
&+ (\bar{\delta}_k - \bar{\delta}_{sp})^T Q_2 (\bar{\delta}_k - \bar{\delta}_{sp}) + (\bar{z}_k - \bar{z}_{sp})^T Q_2 (\bar{z}_k - \bar{z}_{sp}) \\
&+ (\mu_k - \mu_{k-1})^T Q_4 (\mu_k - \mu_{k-1}) + (\mu_k - \mu_{sp})^T Q_1 (\mu_k - \mu_{sp})
\end{aligned} \tag{E.14}$$

Après simplification, on aura l'écriture suivante :

$$\begin{aligned}
\min_{u, \delta, z} J = & (H_{M1}\mu_k)^T Q_5 H_{M1}\mu_k + (H_{M1}\mu_k)^T Q_5 (H_{M2}\delta_k) + (H_{M1}\mu_k)^T Q_5 (H_{M3}z_k) \\
& + (H_{M1}\mu_k)^T Q_5 (\Phi_M x_k) + (H_{M1}\mu_k)^T Q_5 (-\psi_{sp}) \\
& + (H_{M2}\delta_k)^T Q_5 (H_{M1}\mu_k) + (H_{M2}\delta_k)^T Q_5 (H_{M2}\delta_k) + (H_{M2}\delta_k)^T Q_5 (H_{M3}z_k) \\
& + (H_{M2}\delta_k)^T Q_5 (\Phi_M x_k) + (H_{M2}\delta_k)^T Q_5 (-\psi_{sp}) \\
& + (H_{M3}z_k)^T Q_5 (H_{M1}\mu_k) + (H_{M3}z_k)^T Q_5 (H_{M2}\delta_k) + (H_{M3}z_k)^T Q_5 (H_{M3}z_k) \\
& + (H_{M3}z_k)^T Q_5 (\Phi_M x_k) + (H_{M3}z_k)^T Q_5 (-\psi_{sp}) \\
& + (\Phi_M x_k)^T Q_5 (H_{M1}\mu_k) + (\Phi_M x_k)^T Q_5 (H_{M2}\delta_k) + (\Phi_M x_k)^T Q_5 (H_{M3}z_k) \\
& + (\Phi_M x_k)^T Q_5 (\Phi_M x_k) + (\Phi_M x_k)^T Q_5 (-\psi_{sp}) \\
& + (\bar{\delta}_k - \bar{\delta}_{sp})^T Q_2 (\bar{\delta}_k - \bar{\delta}_{sp}) + (\bar{z}_k - \bar{z}_{sp})^T Q_2 (\bar{z}_k - \bar{z}_{sp}) \\
& + (\mu_k - \mu_{k-1})^T Q_4 (\mu_k - \mu_{k-1}) + (\mu_k - \mu_{sp})^T Q_1 (\mu_k - \mu_{sp})
\end{aligned} \tag{E.15}$$

Par le développement des termes sur les vecteurs de commande, on obtient la forme suivante :

$$\begin{aligned}
\min_{u, \delta, z} J = & (H_{M1}\mu_k)^T Q_5 H_{M1}\mu_k + (H_{M1}\mu_k)^T Q_5 (H_{M2}\delta_k) + (H_{M1}\mu_k)^T Q_5 (H_{M3}z_k) \\
& + (H_{M1}\mu_k)^T Q_5 (\Phi_M x_k) + (H_{M1}\mu_k)^T Q_5 (-\psi_{sp}) \\
& + (H_{M2}\delta_k)^T Q_5 (H_{M1}\mu_k) + (H_{M2}\delta_k)^T Q_5 (H_{M2}\delta_k) + (H_{M2}\delta_k)^T Q_5 (H_{M3}z_k) \\
& + (H_{M2}\delta_k)^T Q_5 (\Phi_M x_k) + (H_{M2}\delta_k)^T Q_5 (-\psi_{sp}) \\
& + (H_{M3}z_k)^T Q_5 (H_{M1}\mu_k) + (H_{M3}z_k)^T Q_5 (H_{M2}\delta_k) + (H_{M3}z_k)^T Q_5 (H_{M3}z_k) \\
& + (H_{M3}z_k)^T Q_5 (\Phi_M x_k) + (H_{M3}z_k)^T Q_5 (-\psi_{sp}) \\
& + (\Phi_M x_k)^T Q_5 (H_{M1}\mu_k) + (\Phi_M x_k)^T Q_5 (H_{M2}\delta_k) + (\Phi_M x_k)^T Q_5 (H_{M3}z_k) \\
& + (\Phi_M x_k)^T Q_5 (\Phi_M x_k) + (\Phi_M x_k)^T Q_5 (-\psi_{sp}) \\
& + (-\psi_{sp})^T Q_5 (H_{M1}\mu_k) + (-\psi_{sp})^T Q_5 (H_{M2}\delta_k) + (-\psi_{sp})^T Q_5 (H_{M3}z_k) \\
& + (-\psi_{sp})^T Q_5 (\Phi_M x_k) + (-\psi_{sp})^T Q_5 (-\psi_{sp}) \\
& (\bar{\delta}_k)^T Q_2 (\bar{\delta}_k) + (\bar{\delta}_{sp})^T Q_2 (\bar{\delta}_{sp}) + (\bar{\delta}_k)^T Q_2 (-\bar{\delta}_{sp}) + (-\bar{\delta}_{sp})^T Q_2 (\bar{\delta}_k) \\
& (\bar{z}_k)^T Q_2 (\bar{z}_k) + (\bar{z}_{sp})^T Q_2 (\bar{z}_{sp}) + (\bar{z}_k)^T Q_2 (-\bar{z}_{sp}) + (-\bar{z}_{sp})^T Q_2 (\bar{z}_k) \\
& (\mu_k)^T Q_4 (\mu_k) + (\mu_{k-1})^T Q_4 (\mu_{k-1}) + (\mu_k)^T Q_4 (-\mu_{k-1}) + (-\mu_{k-1})^T Q_4 (\mu_k) \\
& (\mu_k)^T Q_1 (\mu_k) + (\mu_{sp})^T Q_1 (\mu_{sp}) + (\mu_k)^T Q_1 (-\mu_{sp}) + (-\mu_{sp})^T Q_1 (\mu_k)
\end{aligned} \tag{E.16}$$

Pour réécrire le critère sous forme compacte quadratique mixte, on introduit un arrangement avec élimination des termes constants apparente, de la forme suivante :





$$E_{M1}\mu_k + E_{M2}\bar{\delta}_k + E_{M3}\bar{z}_k \leq E_{M4}x_k + E_{M5} \quad (\text{E.22})$$

Cette forme peut s'écrire de la façon suivante :

$$\begin{bmatrix} E_{M1} & E_{M2} & E_{M3} \end{bmatrix} \begin{bmatrix} \mu_k \\ \bar{\delta}_k \\ \bar{z}_k \end{bmatrix} \leq E_{M4}x_k + E_{M5} \quad (\text{E.23})$$

On pose :  $S_1 = [E_{M1} \ E_{M2} \ E_{M3}]$ .

$$E_{M1} = \begin{bmatrix} -E_1 & 0 & \dots & & & & 0 \\ -E_4B_1 & -E_1 & \dots & & & & 0 \\ -E_4A^2B_1 & -E_4B_1 & \dots & & & & 0 \\ \vdots & \vdots & & -E_1 & & & \vdots \\ -E_4A^{(m-2)}B_1 & -E_4A^{(m-3)}B_1 & \dots & -E_4B_1 & & & -E_1 \\ -E_4A^{(m-1)}B_1 & -E_4A^{(m-2)}B_1 & \dots & -E_4AB_1 & & & -E_1 - E_4B_1 \\ -E_4A^mB_1 & -E_4A^{(m-1)}B_1 & \dots & -E_4A^2B_1 & & & -E_1 - E_4B_1 - E_4AB_1 \\ & \vdots & & \vdots & & & \vdots \\ -E_4A^{(p-2)}B_1 & -E_4A^{(p-3)}B_1 & \dots & -E_4A^{(p-m)}B_1 & & & -E_1 + \sum_{i=0}^{p-m-1} -E_4A^iB_1 \end{bmatrix}$$

$$E_{Mi} = \begin{bmatrix} E_i & 0 & \dots & \dots & & & 0 \\ -E_4B_i & E_i & \dots & \dots & & & 0 \\ -E_4AB_i & -E_4B_i & \dots & \dots & & & 0 \\ \vdots & \vdots & \ddots & & & & \vdots \\ -E_4A^{(p-3)}B_i & -E_4A^{(p-4)}B_i & \dots & \dots & E_i & & 0 \\ -E_4A^{(p-2)}B_i & -E_4A^{(p-3)}B_i & \dots & \dots & -E_4B_i & E_i & \end{bmatrix} \quad \text{whre } i=2,3 \quad E_{M5} = \begin{bmatrix} -E_5 \\ -E_5 \\ -E_5 \\ \vdots \\ p \text{ times} \\ \vdots \\ -E_5 \end{bmatrix}$$

D'autre part, on ajoute la contrainte sur la sortie  $y_k$  sur un horizon  $p$  :

$$\begin{cases} \psi_k = H_{M1}\mu_k + H_{M2}\bar{\delta}_k + H_{M3}\bar{z}_k + \Phi_M\bar{x}_k \leq \psi_{max} \\ -\psi_k = -H_{M1}\mu_k - H_{M2}\bar{\delta}_k - H_{M3}\bar{z}_k - \Phi_M\bar{x}_k \leq -\psi_{min} \end{cases} \quad (\text{E.24})$$

On réécrit l'équation (E.24), en fonction du vecteur  $\chi$  :

$$\begin{bmatrix} H_{M1} & H_{M2} & H_{M3} \\ -H_{M1} & -H_{M2} & -H_{M3} \end{bmatrix} \begin{bmatrix} \mu_k \\ \bar{\delta}_k \\ \bar{z}_k \end{bmatrix} \leq \begin{bmatrix} \psi_{max} - \Phi_M\bar{x}_k \\ -\psi_{min} + \Phi_M\bar{x}_k \end{bmatrix} \quad (\text{E.24})$$

On pose :  $S_2 = [H_{M1} \ H_{M2} \ H_{M3}]$ .

Enfin, dans cette partie, on formule le vecteur des contraintes sur les commandes continues.

$$\begin{cases} u \leq u_{max} \\ -u \leq -u_{min} \end{cases} \quad (E.25)$$

Pour un horizon de commande  $m$  à l'instant  $k$ , on obtient :

$$\begin{cases} \mu_k \leq \mu_{max} \\ -\mu_k \leq -\mu_{min} \end{cases} \quad (E.26)$$

Pour pouvoir utiliser cette contrainte dans l'équation d'inégalité, on reformule (E.26) :

$$\begin{bmatrix} S_3 \\ -S_3 \end{bmatrix} \begin{bmatrix} \mu_k \\ \bar{\delta}_k \\ \bar{z}_k \end{bmatrix} \leq \begin{bmatrix} \mu_{max} \\ -\mu_{min} \end{bmatrix} \quad (E.27)$$

Où :

$$S_3 = [\text{eye}(m(\text{size}(B_1, 2))) \quad \text{zeros}(\text{size}(B_1, 2)), p(\text{size}(B_2, 2))) \quad \text{zeros}(m(\text{size}(B_1, 2)), p(\text{size}(B_3, 2)))]$$

Finalement, on obtient la matrice et le vecteur de l'inégalité décrivant les contraintes :

$$A_{in} = \begin{bmatrix} S_1 \\ S_2 \\ -S_2 \\ S_3 \\ -S_3 \end{bmatrix}, \quad \begin{cases} b_1 = E_{M5} - E_{M4}x_k \\ b_{21} = \psi_{max} - \Phi_M x_k - S_d d_k \\ b_{22} = -\psi_{max} + \Phi_M x_k + S_d d_k \\ b_{31} = \mu_{max} \\ b_{32} = -\mu_{min} \end{cases} \Rightarrow b_{in} = \begin{bmatrix} b_1 \\ b_{21} \\ b_{22} \\ b_{31} \\ b_{32} \end{bmatrix} \quad (E.27)$$

# Références Bibliographiques

- [1] J. Thomas "Estimation et commande prédictive à horizon glissant de systèmes hybrides", thèse de doctorat, de l'université Paris XI Orsay, Supelec, 2004.
- [2] P. Riedinger "Contribution à la commande optimale des systèmes dynamiques hybrides", Centre de recherche en Automatique de Nancy & Institut National Polytechnique de Lorraine, 2000.
- [3] A. Bemporad and M. Morari "Control of Systems Integrating Logic, Dynamics and Constraints", In *Automatica* vol. 35, no. 3, pp. 407–427, March 1999.
- [4] N.N. Nareshkumar "A Multiple model approach for modeling, identification and control of nonlinear hybrid systems", Indian Institute of technology Bombay, 2009.
- [5] R. Hakiki "Etude et Analyse des systèmes dynamiques hybrides : Approche par les réseaux de Petri hybrides et Automates hybrides linéaires", Université d'Oran, 2010.
- [6] F. D. Torrisi and A. Bemporad "HYSDEL - A Tool for Generating Computational Hybrid Models for Analysis and Synthesis Problems", In *IEEE Transactions on Control Systems Technology*, vol. 12, no. 2, pp. 235–249, March 2004.
- [7] D. M. W. Leenaerts and W.M.G. van Bokhoven "Piecewise Linear Modeling and Analysis", Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.
- [8] W. Heemels, J. M. Schumacher and S. Weiland "Linear Complementarity Systems", In *SIAM Journal of Applied Mathematics*, 2000.
- [9] B. De Schutter and B. de Moor "The Extended Linear Complementarity Problem and the modeling and Analysis of Hybrid Systems. In P. Antsaklis, W. Kohn, M. Lemmon, A. Nerode, and S. Sastry, editors, *Hybrid Systems V*, volume 1567 of *Lecture Notes in Computer Science*, pages 70–85. Springer, 1999.
- [10] De Schutter, B. and T. van den Boom "On Model Predictive Control for MaxMin-Plus-Scaling Discrete Event Systems", Technical Report, The Netherlands, February 2000.
- [11] D.ED Belkhiat "Diagnostic d'une classe de systèmes linéaires à commutations : Approche à base d'observateurs robustes", Thèse de doctorat, Université de Reims Champagne Ardenne, France, 2011.
- [12] A. Levis and al "Challenge to control a collective view", *IEEE transaction on Automatic Control*, pp. 274-285, 1987.
- [13] Antsaklis et al "Hybrid system modelling and event identification" Technical report of the ISIS Group at the University of Notre Dame ISIS-93-002, Notre Dame, IN, USA, 1993.
- [14] Z. Li, Y. Soh and C. Wen "Switched impulsive systems: Analysis, Design, and applications", *Lecture Notes in Control and Information Sciences* vol. 313, Springer, New York, 2005.

- [15] M.S Branicky "Studies in hybrid systems Modeling, Analysis and Control", thesis for Doctor of Science, Massachusettes Institute of Technology, USA, 1995.
- [16] A. Bemporad, D. Mignone et M. Morari "Moving horizon estimation for hybrid systems and fault detection", In Proceedings of the American Control Conference, San Diego, 1999.
- [17] C. Tomlin "Conflict resolution for air traffic management" - Proceedings of the. Conference on Decision & Control, New Orleans, LA - December 1995.
- [18] J. Lygeros et N. Lynch "String of vehicles: modeling and safety conditions", in T. Henzinger and S. Sastry (eds). Hybrid Systems: Computation and Control (HSCC), Proceedings of the 1st International Workshop on Hybrid Systems, number 1386 in Lecture Notes in Computer Science, Springer-Verlag, Berkeley, CA, USA, 1998.
- [19] W. Colmenares et al "MLD systems: Modeling and Control. Experience with a pilot process", in: Proceedings of the 2001 IEEE International Conference on Control Applications (CCA'01), 5-7 Sept. 2001, IEEE, Mexico City, Mexico, 2001, pp. 618.
- [20] B. Potocnik et al "Hybrid Modelling and Optimal Control of a Multiproduct Batch Plant", Control Engineering Practice, 2004.
- [21] A. Bemporad, F.D. Torrisi, M. Morari "Discrete-time Hybrid Modeling and Verification of the Batch Evaporator Process Benchmark", European Journal of Control, 2001.
- [22] G. Ferrari-Trecate, D. Mignone, D. Castagnoli, M. Morari "Mixed Logic Dynamical Model of a Hydroelectric Power Plant", in, Automatic Control Lab, Swiss Federal Institute of Technology (ETH), Zurich, 2000.
- [23] M.E. van Wissen, J. Smeets, A. Muller, P.J.T. Verheijen "Discrete event modelling and dynamic optimisation of a sugar plant", in: FOCAPO, Florida, USA, 2003.
- [24] B. De Schutter "Optimal control of a class of linear hybrid systems with saturation", SIAM Journal on Control and Optimization, 2000.
- [25] A.J. Van der Schaft and J. M. Schumacher "An Introduction to Hybrid Dynamical Systems", Volume 251 of Lecture Notes in Control and Information Sciences, Springer, 2000.
- [26] P.J Mosterman "An overview of hybrid simulation phenomena and their support by simulation packages. In: F. W. Vaandrager, & J. H. Van Schuppen. Hybrid systems: computation and control. Lecture Notes in Computer Science 1569. (pp. 165–177) Berlin: Springer, 1999.
- [27] T. Henzinger "The theory of hybrid automata", Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science, pp. 278-292, 1996.
- [28] R., C. Alur et al "The algorithmic analysis of hybrid systems", Theoretical Computer Science, vol. 138, pp. 3-34, 1995.
- [29] A. Van der Schaft et J. Schumacher "Complementarity modeling of hybrid systems", IEEE Transactions on Automatic Control 43(4), 483-490, 1998.

- [30] D. Mignone "Control and Estimation of Hybrid Systems with Mathematical Optimization", Swiss Federal Institute of Technology (ETH), thesis for the degree of Dr. Science, Italy, 2002.
- [31] E.D. Sontag "Nonlinear regulation: the piecewise linear approach", In IEEE Transactions on Automatic Control, 26(2):346-358, 1981.
- [32] E.D. Sontag "Interconnected Automata and Linear Systems", A Theoretical Framework in Discrete-Time. Hybrid Systems III – Verification and Control, no. 1066 in LNCS, pp. 436-448, Springer-Verlag, 1996.
- [33] D. Mignone, G. Ferrari-Trecate and M. Morari. "Stability and Stabilization of Piecewise Affine and Hybrid Systems: An LMI Approach", In 39th IEEE Conference on Decision and Control, December, pp. 504-509, 2000.
- [34] G. Feng "Stability Analysis of Piecewise Discrete-Time Linear Systems", IEEE Transactions on Automatic Control, vol. 47, no. 7, pp. 1108–1112, 2002.
- [35] A. Bemporad and M. Morari "Robust Model Predictive Control: A Survey", In Garulli, A., A. Tesi and A. Vincio (editors): Robustness in Identification and control, volume 25 of Lecture Notes in Control and Information Sciences, pp. 207-226. Springer-Verlag, 1999.
- [36] A. Bemporad, F. D. Torrisi and M. Morari "Optimization-Based Verification and Stability Characterization of Piecewise Affine and Hybrid systems" In B. Krogh and N. Lynch, editors, Hybrid Systems: Computation and Control, volume 1790 of Lecture Notes in Computer Science, pp. 45–58. Springer Verlag, 2000.
- [37] A. Bemporad, G. Ferrari-Trecate and M. Morari "Observability and Controllability of Piecewise Affine and Hybrid Systems", In IEEE Transactions on Automatic Control, vol. 45, no. 10, pp. 1864–1876, 2000.
- [38] W. P. M. H., Heemels, B. De Schutter and A. Bemporad "Equivalence of Hybrid Dynamical Models", In Automatica, vol. 37, no. 7, pp. 1085-1091, July 2001.
- [39] F. D. Torrisi and A. Bemporad: "HYSDEL -A Tool for Generating Computational Hybrid Models for Analysis and Synthesis Problems", In IEEE Transactions on Control Systems Technology, vol. 12, no. 2, pp. 235–249, March 2004.
- [40] M. Kvasnica, P. Grieder, M. Baotić and M. Morari "Multi Parametric Toolbox (MPT) ", In Alur R., G. Pappas (editors) : Hybrid Systems: Computation and Control, volume 2993 of Lecture Notes in Computer Science, pp.448-462. Springer-Verlag, 2004.
- [41] P. Mosterman "An overview of hybrid simulation phenomena and their support by simulation packages", Hybrid systems: computation and control. Lecture Notes in Computer Science 1569. (pp. 165–177) Berlin: Springer.
- [42] Jan Lunze, Françoise Lamnabhi-Lagarrigue "Handbook of Hybrid Systems Control : Theory, Tools, Applications", 2009.

- [43] Flaus "Modular Representation Of Complex Hybrid Systems Application To The Simulation Of Batch Processes", Simulation Practice and theory (SIMPRA), Laboratoire d'Automatique de Grenoble, 2000.
- [44] Mosterman, "HYBRISIM - a modeling and simulation environment for hybrid bond graphs", (2002).
- [45] H. Taylor "A Modeling Language for Hybrid systems", Odyssey Research Associates, Symposium IEEE/IFAC on Computer - Aided Control System Design, Tucson, March 1994.
- [46] M. Daigle, I. Roychoudhury, Gautam Biswas, and Xenofon Koutsoukos "Efficient Simulation of Component-Based Hybrid Models Represented as Hybrid Bond Graphs", HSCC-Italy, Vanderbilt University, USA, 2007.
- [47] P. Schwarz "Simulation of systems with dynamically varying model structure", Mathematics and Computers in Simulation, pp 850-863, ScienceDirect, 2008.
- [48] B. Abderrahim "Modélisation d'une boucle à eau pressurisée Simulation d'un accident de perte du réfrigérant : Utilisation du code RELAP/MOD1", Centre de Développement des Systèmes Energétiques, Thèse de Magister en Génie Nucléaire, 1991.
- [49] A. HADJAM "Modélisation et simulation d'un accident de type LOCA sur une boucle pressurisée de type PWR", Thèse de Magister en Génie Nucléaire, 2007.
- [50] Apros – Dynamic Process Simulation Software, <[www.apros.fi](http://www.apros.fi)>.
- [51] H.P. Williams "Model Building in Mathematical Programming" 3<sup>rd</sup> ed., New York : Wiley, 2013.
- [52] T.M. Cavalier, P. M Pardalos et A. L. Soyster "Modeling and integer programming techniques applied to propositional calculus", Comput. Oper. Res., 17(6), 561-570, 1990.
- [53] R. Raman, I. E. Grossmann "Integration of logic and heuristic knowledge in MINLP optimization for process synthesis", Computers Chem. Engineering, 16(3), 155-171, 1992.
- [54] A. Schrijver "Theory of Linear and Integer Programming", Wiley-Interscience, 1986.
- [55] J. Richalet, A. Rault, J. L. Testud et J. Jap "Model predictive heuristic control : application to industrial processes", Automatica, 14(5), pp. 413-428, 1978.
- [56] D.W. Clarke, C. Mohtadi et P. S. Tuffs, 1987 "Generalized predictive control – Part I. and II", Automatica, 23(2), pp. 137-160, 1987.
- [57] E.F. Camacho et C. Bordons "Model predictive control", Springer-Verlag, London, 1999.
- [58] R. Fletcher and S. Leyffer "Numerical experience with lower bounds for MIQP branch and bound", 1995.
- [59] E.N. Pistikopoulos, V. Dua, N.A. Bozinis, A. Bemporad et M. Morari "On-line Optimization via Off-line Parametric Optimization Tools", Published in Computers & Chemical Engineering, Vol. 24, No. 2-7, pp. 183-188, 2000.

- [60] V. Dua, N.A. Bozinis et E.N. Pistikopoulos "A multiparametric programming approach for mixed-integer quadratic engineering problems", Computers and Chemical Engineering 26, 715-733.
- [61] F. Borrelli "Constrained Optimal Control of Linear and Hybrid Systems", Lecture Notes in Control and Information Sciences.290.Springer-Verlag, 2003.
- [62] M. Kvasnica, P. Grieder, M. Baotic et M. Morari "Multi-Parametric Toolbox (MPT).Hybrid Systems: Computation and Control, Volume 2993 of Lecture Notes in Computer Science, pp. 448-462, 2004.

## **Production scientifique dans le cadre des travaux de Magister**

Le travail présenté dans ce mémoire a été valorisé par plusieurs conférences internationales :

M.F. Belazreg, K. Halbaoui and D. Boukhetala « Model Predictive Control of Hybrid System using MLD Formalism: Application to the Three-Tank Benchmark», The Second International Conference on Electrical Engineering and Control Applications (**ICEECA2014**), November 18-20, 2014, Constantine, Algeria (accepted).

M.F. Belazreg, K. Halbaoui and D. Boukhetala « Modelling, Simulations and Control of Industrial Process with APROS Software », first International Conference on Materials for Energy & Environmental Engineering (**ICM3E'14**) 23 - 25 November 2014 – Alger (accepted).

M.F. Belazreg, K. Halbaoui and D. Boukhetala « Modelling and Predictive Control of Hybrid Systems Using APROS Software: Application to the Three-Tank Benchmark Problem », **APROS@User Group Seminar** 2014, Finland, Espoo, 20-23 October 2014 (presented).

M.F. Belazreg, K. Halbaoui and D. Boukhetala «Modelling, Simulation and Control of Hybrid Systems Using APROS Software», eighth African Conference on Nuclear Research Reactor Utilization and Safety, **COMENA**, du 14 au 16 December 2014 (presented).

M.F. Belazreg, K. Halbaoui and D. Boukhetala « Model Predictive Control of Hybrid System using MLD Formalism: Application to the Three-Tank Benchmark », The Eighth International Conference on Electrical Engineering (**CEE'2014**), November 16-17, 2014, Batna, Algeria (accepted).