



HAL
open science

Commande prédictive des systèmes PWA avec entrées mixtes

Sylvain Leirens, Jean Buisson

► **To cite this version:**

Sylvain Leirens, Jean Buisson. Commande prédictive des systèmes PWA avec entrées mixtes. CIFA 2006: Conférence Internationale Francophone d'Automatique, 2006, Bordeaux, France. pp.CDROM. hal-00114726

HAL Id: hal-00114726

<https://hal.science/hal-00114726v1>

Submitted on 10 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Commande prédictive des systèmes PWA avec entrées mixtes

Sylvain LEIRENS, Jean BUISSON

Équipe Automatique des Systèmes Hybrides
Supélec–IETR, av. de la Boulaie, BP 81127, 35511 Cesson-Sévigné Cedex, France

sylvain.leirens@supelec.fr
<http://www.rennes.supelec.fr/ren/rd/ash/>

Résumé— Cet article présente un algorithme d'optimisation mixte dédié à la commande prédictive des systèmes affines par morceaux (PWA) avec entrées mixtes (*i.e.* continues et discrètes). En exploitant la structure particulière du problème d'optimisation, le nombre et la dimension des sous-problèmes à résoudre pour trouver l'optimum sont réduits.

Cette approche est appliquée à un cas d'étude classique de l'automatique hybride : le contrôle des niveaux de liquide d'un système à trois cuves.

Mots-clés— commande prédictive, systèmes hybrides, systèmes PWA, optimisation mixte.

I. INTRODUCTION

Une grande part de la littérature concernant l'automatique *avancée* implique le concept d'optimalité et nécessite des méthodes d'optimisation adaptées. Pour les problèmes de commande des procédés industriels complexes et sous contraintes, la commande prédictive est devenue un standard, bien qu'elle requiert une procédure d'optimisation effectuée en ligne. À chaque instant d'échantillonnage, une séquence de commandes optimale (vis à vis d'un critère donné) sur un horizon de prédiction fini est trouvée, dont seule la première commande est appliquée au système. Cette procédure est répétée à l'instant suivant (horizon *glissant*).

Les systèmes hybrides sont des systèmes dynamiques qui impliquent la présence de variables continues et discrètes dans les modèles. En conséquence, les méthodes d'optimisation requises par la commande prédictive sont confrontées à des aspects combinatoires. Parmi les différents paradigmes permettant de construire des modèles hybrides, certaines classes de systèmes se prêtent particulièrement à la mise en œuvre de lois de commande prédictive, notamment dans leur formulation à temps discret. On distingue la classe des systèmes affines par morceaux (PWA pour PieceWise Affine) qui sont définis par des dynamiques affines sur des partitions polyédrales de l'espace d'état-commande (continu ou mixte). L'effort de recherche concerne principalement des techniques de réduction de la complexité combinatoire du problème d'optimisation associé à la commande prédictive afin de ne pas avoir à énumérer toutes les possibilités pour la commande sur l'horizon de prédiction, ce qui engendre un (très) grand nombre de sous-problèmes à résoudre pour trouver l'optimum à chaque instant d'échantillonnage. Pour les systèmes PWA continus (*i.e.* pas d'état ni d'entrées discrets), les concepts d'atteignabilité et de

graphe de transition d'état sont utilisés dans [1] afin de proposer un algorithme qui réduit le nombre de ces sous-problèmes. Pour s'affranchir du besoin de calcul *en ligne* requis par la commande prédictive, la résolution du problème de commande peut être effectuée *hors ligne* en utilisant la programmation multi-paramétrique [2]. Toutefois cette dernière approche se limite à des systèmes de très petite dimension. Pour les multimodèles affines (SAS pour Switched Affine Systems) qui sont une sous-classe des systèmes PWA impliquant des entrées de commande mixtes, mais pas de partitionnement de l'espace d'état-commande continu, un algorithme d'énumération partielle est proposé dans [3] pour en réduire la dimension et le nombre. C'est un algorithme de type branch & bound (B&B) qui exploite les caractéristiques particulières du problème de commande prédictive.

On distingue aussi le formalisme Mixed Logical Dynamical (MLD), dont l'équivalence avec les systèmes PWA a été prouvée dans [4], qui aboutit à un modèle composé d'équations d'état linéaires à temps discret et d'un ensemble d'inégalités linéaires impliquant des variables réelles et binaires [5]. La mise en œuvre de la commande prédictive pour les systèmes hybrides modélisés sous forme MLD requiert la résolution de problèmes d'optimisation standards qui sont mixtes linéaires ou quadratiques, appelés MILP/MIQP (pour Mixed Integer Linear/Quadratic Programming). Ces problèmes d'optimisation difficiles sont résolus en utilisant les techniques du B&B et requièrent une capacité de calcul importante. Différentes approches ont été proposées pour réduire la complexité liée à l'utilisation du formalisme MLD. Dans [6], les formalismes PWA et MLD sont combinés pour aboutir au formalisme SPMLD (State Partition Based MLD) qui consiste à utiliser un modèle MLD particulier dans chaque partition de l'espace d'état. Dans [7], un outil de test de satisfaisabilité des contraintes logiques (SAT) est combiné à un algorithme de B&B pour diminuer la complexité combinatoire.

Dans cet article, l'approche proposée dans [3], qui consiste en un algorithme d'énumération partielle exploitant les caractéristiques du problème, est étendue aux systèmes PWA (avec entrées mixtes). Il est organisé comme suit. Les systèmes PWA sont présentés en section II. Le principe de la commande prédictive (hybride) est introduit en section III et la section IV est consacrée au problème d'optimisation mixte associé. La section V présente la mise

en œuvre de l'approche proposée et les résultats de simulation obtenus pour le contrôle des niveaux de liquide d'un cas d'étude classique de l'automatique hybride : le système à trois cuves. Les conclusions sont données en section VI.

II. SYSTÈMES PWA

La classe de systèmes, appelés systèmes PWA avec entrées mixtes (mais dont l'état est continu), à laquelle on s'intéresse dans cet article est définie par les équations à temps discret suivantes :

$$\mathbf{x}(k+1) = \mathbf{A}_i \mathbf{x}(k) + \mathbf{B}_i \mathbf{u}_c(k) + \mathbf{a}_i \quad (1a)$$

$$\mathbf{y}(k) = \mathbf{C}_i \mathbf{x}(k) + \mathbf{D}_i \mathbf{u}_c(k) + \mathbf{c}_i \quad (1b)$$

où $\mathbf{x} \in \mathbb{X} \subset \mathbb{R}^{n_x}$ est le vecteur d'état, $\mathbf{u}_c \in \mathbb{U} \subset \mathbb{R}^{n_{uc}}$ est le vecteur des entrées continues et $\mathbf{y} \in \mathbb{Y} \subset \mathbb{R}^{n_y}$ est le vecteur de sortie du système. $\mathbf{A}_i \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B}_i \in \mathbb{R}^{n_x \times n_{uc}}$, $\mathbf{C}_i \in \mathbb{R}^{n_y \times n_x}$, $\mathbf{D}_i \in \mathbb{R}^{n_y \times n_{uc}}$, $\mathbf{a}_i \in \mathbb{R}^{n_x}$ et $\mathbf{c}_i \in \mathbb{R}^{n_y}$. L'indice i dénote le mode $i(k) \in \mathbb{I} \subset \mathbb{N}$ à l'instant k qui est défini par une valeur du vecteur des entrées discrètes $\mathbf{u}_d(k)$ ($\mathbf{u}_d \in \mathbb{U}_d$ avec par ex., $\mathbb{U}_d \subset \mathbb{Z}^{n_{ud}}$) et l'appartenance à une partition χ_j ($j \in \mathbb{J} \subset \mathbb{N}$) de l'espace d'état-commande continu, indépendante des entrées discrètes \mathbf{u}_d :

$$(\mathbf{x}(k), \mathbf{u}_c(k)) \in \chi_j = \{(\mathbf{x}, \mathbf{u}_c) \mid \mathbf{F}_j \mathbf{x} + \mathbf{G}_j \mathbf{u}_c \leq \mathbf{f}_j\} \quad (2)$$

On définit alors une fonction $\varphi : \mathbb{X} \times \mathbb{U}_c \times \mathbb{U}_d \rightarrow \mathbb{I}$ telle que :

$$i(k) = \varphi(\mathbf{x}(k), \mathbf{u}_c(k), \mathbf{u}_d(k)) \quad (3)$$

Pour faire le lien avec les autres formalismes hybrides, le mode dans lequel se trouve le système est analogue à l'état discret (d'un automate hybride par ex.). L'entrée de commande \mathbf{u} du système se compose des entrées continues (\mathbf{u}_c) et discrètes (\mathbf{u}_d) avec $n_u = n_{uc} + n_{ud}$. Le nombre total de modes pour les systèmes est donné par :

$$p = \text{card}(\mathbb{I}) = \text{card}(\mathbb{J}) \times \text{card}(\mathbb{U}_d) \quad (4)$$

III. COMMANDE PRÉDICTIVE

Le principe de la commande prédictive consiste à résoudre un problème de commande optimale en boucle ouverte sur un horizon de prédiction glissant. La boucle est fermée par l'utilisation d'une nouvelle mesure de l'état du système à chaque instant d'échantillonnage. Le modèle (PWA) de prédiction est stationnaire, on peut donc utiliser l'origine des temps comme instant courant sans perte de généralité dans les équations. Soit \mathbf{U} la séquence de commandes sur l'horizon de prédiction, choisi de longueur N :

$$\mathbf{U} = (\mathbf{u}^T(0) \ \mathbf{u}^T(1) \ \dots \ \mathbf{u}^T(N-1))^T \quad (5)$$

avec $\mathbf{u} = (\mathbf{u}_c^T \ \mathbf{u}_d^T)^T$ et soit la fonction de coût suivante :

$$J_N(\mathbf{x}(0), \mathbf{U}) = F(\mathbf{x}(N)) + \sum_{k=0}^{N-1} L(\mathbf{x}(k), \mathbf{u}(k)) \quad (6)$$

En pratique, la fonction de coût (ou critère à optimiser) inclut un terme basé sur les écarts de la sortie \mathbf{y} par rapport à une référence et un terme basé sur l'entrée de commande \mathbf{u} . Soient :

$$L(\mathbf{x}(k), \mathbf{u}(k)) = \|\mathbf{y}(k) - \mathbf{y}_r\|_{\mathbf{Q}_y} + \|\mathbf{u}(k)\|_{\mathbf{Q}_u} \quad (7a)$$

$$F(\mathbf{x}(N)) = \|\mathbf{x}(N) - \mathbf{x}_r\|_{\mathbf{Q}_f} \quad (7b)$$

où \mathbf{y}_r et \mathbf{x}_r sont respectivement les références pour la sortie et l'état (final). En utilisant la norme 2, la fonction de coût est quadratique, *i.e.* $\|\mathbf{w}\|_{\mathbf{Q}} \triangleq \mathbf{w}^T \mathbf{Q} \mathbf{w}$. Les matrices de pondération sont telles que $\mathbf{Q}_f \geq 0$, $\mathbf{Q}_y \geq 0$ et $\mathbf{Q}_u > 0$ (si des contraintes sur \mathbf{u} sont présentes, typiquement des contraintes sur les actionneurs, il suffit que la matrice \mathbf{Q}_u soit semi-définie positive puisque \mathbf{u} ne peut pas avoir de composantes infinies). On peut ajouter un terme sur l'état $\|\mathbf{x}(k+1) - \mathbf{x}_r\|_{\mathbf{Q}_x}$ avec $\mathbf{Q}_x \geq 0$.

À chaque instant d'échantillonnage, le problème d'optimisation \mathcal{P}_N suivant doit être résolu, l'exposant o signifiant l'optimalité :

$$\mathcal{P}_N(\mathbf{x}(0)) : J_N^o(\mathbf{x}(0)) = \min_{\mathbf{U}} J_N(\mathbf{x}(0), \mathbf{U}) \quad (8)$$

sous les contraintes du modèle (1a)–(2). Une contrainte égalité sur l'état final $\mathbf{x}(N) = \mathbf{x}_f$ peut être ajoutée, notamment pour garantir la stabilité du système bouclé [8].

Une particularité du problème d'optimisation ainsi formulé est sa nature mixte. La présence d'entrées de commande continues (\mathbf{u}_c) et discrètes (\mathbf{u}_d) conduit à considérer la séquence de commandes recherchée \mathbf{U} comme une séquence de commandes continues, notée \mathbf{U}_c , et une séquence de commandes discrètes, notée \mathbf{U}_d , sur l'horizon de prédiction N . On a donc :

$$J_N^o(\mathbf{x}(0)) = \min_{\mathbf{U}_d} \left(\min_{\mathbf{U}_c} J_N(\mathbf{x}(0), (\mathbf{U}_c, \mathbf{U}_d)) \right) \quad (9)$$

sous les contraintes du modèle, pour $k = 0, 1, \dots, N-1$:

$$\mathbf{x}(k+1) = \mathbf{A}_i \mathbf{x}(k) + \mathbf{B}_i \mathbf{u}_c(k) + \mathbf{a}_i \quad (10a)$$

$$\mathbf{y}(k) = \mathbf{C}_i \mathbf{x}(k) + \mathbf{D}_i \mathbf{u}_c(k) + \mathbf{c}_i \quad (10b)$$

$$\mathbf{F}_i \mathbf{x}(k) + \mathbf{G}_i \mathbf{u}_c(k) \leq \mathbf{f}_i \quad (10c)$$

avec $i(k) = \varphi(\mathbf{x}(k), \mathbf{u}_c(k), \mathbf{u}_d(k))$.

On note $\mathbf{I} = (i(0) \ i(1) \ \dots \ i(N-1))^T \in \mathbb{I}^N$ une séquence de modes sur l'horizon N . Elle caractérise une séquence \mathbf{U}_d et N ensembles de contraintes (10a)–(10c) pour $k = 0, 1, \dots, N-1$. Le problème \mathcal{P}_N peut se reformuler ainsi :

$$J_N^o(\mathbf{x}(0)) = \min_{\mathbf{I}} \left(\min_{\mathbf{U}_c} J_N(\mathbf{x}(0), (\mathbf{U}_c, \mathbf{I})) \right) \quad (11)$$

Pour une séquence de modes donnée \mathbf{I} , le coût :

$$J_N^*(\mathbf{x}(0), \mathbf{I}) = \min_{\mathbf{U}_c} J_N(\mathbf{x}(0), (\mathbf{U}_c, \mathbf{I})) \quad (12)$$

est le coût optimal trouvé en résolvant un sous-problème d'optimisation continu (la séquence \mathbf{I} – donc \mathbf{U}_d – est fixée) sous contraintes. Ce sous-problème peut être transformé en un problème standard de programmation quadratique (QP) moyennant quelques étapes de calcul supplémentaires. L'exposant $*$ signifie l'optimalité vis à vis d'une séquence de commandes discrètes donnée. Ainsi, la séquence de commandes continues notée \mathbf{U}_c^* est dite optimale en regard de la séquence de commandes discrètes \mathbf{U}_d donnée. Toutefois, le problème d'optimisation sous contraintes (12) n'est pas nécessairement faisable, *i.e.* pour la séquence de commandes discrètes donnée, aucune solution ne satisfait les contraintes du modèle (10a)–(10c).

IV. OPTIMISATION MIXTE

Le problème d'optimisation associé à la commande prédictive des systèmes PWA étant formulé, on s'intéresse dans cette section à sa résolution.

A. Énumération exhaustive

La méthode la plus simple pour trouver l'optimum recherché consiste à énumérer toutes les séquences de modes possibles sur l'horizon de prédiction puis à résoudre les sous-problèmes QP associés aux séquences correspondantes. Pour une séquence de modes donnée, deux cas de figure peuvent se présenter :

- il n'y a pas de solution satisfaisant les contraintes, le sous-problème QP est infaisable ;
- le sous-problème QP est faisable, une séquence continue optimale est trouvée.

Pour chaque séquence de modes \mathbf{I} où le sous-problème QP correspondant est faisable, une séquence continue optimale \mathbf{U}_c^* en regard de la séquence discrète \mathbf{U}_d associée à \mathbf{I} et une valeur de la fonction de coût (12) sont obtenues. L'optimum recherché est donc donné par la séquence de modes qui minimise (12) :

$$J_N^o(\mathbf{x}(k)) = \min_{\mathbf{I}} (J_N^*(\mathbf{x}(k), \mathbf{I})) \quad (13)$$

Simple en apparence, cette méthode, que l'on qualifie d'énumération exhaustive, est rapidement mise en défaut lorsque le nombre de modes et/ou la longueur de l'horizon de prédiction augmentent. C'est un phénomène d'explosion combinatoire : le nombre de possibilités à énumérer croît exponentiellement avec le nombre de modes et la longueur de l'horizon. Soit p le nombre de modes possibles pour le système. Une énumération exhaustive des séquences de modes conduit à considérer p^N sous-problèmes QP.

La figure 1 est une représentation de l'arbre des possibilités : chaque feuille (symbolisée par un point noir) est un sous-problème QP à résoudre et l'une d'elle est l'optimum recherché. La profondeur dans l'arbre croît avec l'horizon de prédiction et à une profondeur donnée, sa largeur est fixée par le nombre de modes possibles du système.

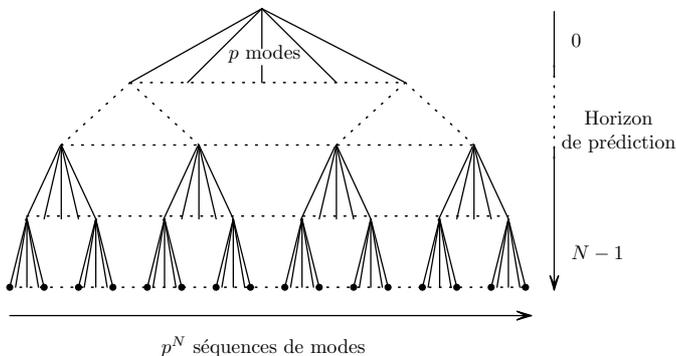


Fig. 1. Énumération exhaustive

Tous les sous-problèmes QP sont de dimension identique, *i.e.* la dimension du vecteur d'optimisation est $\dim(\mathbf{U}_c) = n_{uc} \times N$.

B. Énumération partielle

L'énumération exhaustive consiste à parcourir la totalité de l'arbre des possibilités de la figure 1. Le problème d'optimisation associé à la commande prédictive a une structure particulière : le coût est additif (les fonctions L et F sont définies positives). L'idée clé de l'algorithme d'énumération partielle proposé s'appuie sur la représentation de l'arbre des possibilités de la figure 1 : connaissant un sous-optimum de (9)–(10c), évaluer des coûts partiels afin d'*élaguer* l'arbre, *i.e.* couper les branches qui ne peuvent pas conduire à l'optimum.

La méthode se compose d'une stratégie de descente pour parcourir l'arbre des possibilités et d'un critère d'élimination de branches.

Stratégie de descente : supposons être P ($< N$) instants d'échantillonnage dans le futur, *i.e.* à une profondeur P dans l'arbre des possibilités. La stratégie proposée est de type *meilleur d'abord* :

- calculer les coûts optimaux J_{P+1} correspondant aux sous-problèmes faisables pour les différents choix possibles du mode ;
- choisir la branche donnant le coût minimal sur l'horizon $P + 1$ pour continuer l'exploration.

Élimination de branches : supposons connaître un premier sous-optimum (toutes les feuilles de l'arbre pour lesquelles le sous-problème QP associé est faisable sont des sous-optima, l'optimum recherché est le meilleur sous-optimum). Parcourir l'arbre en éliminant les branches pour lesquelles :

- soit le coût optimal sur un horizon réduit est supérieur au coût du sous-optimum connu ;
- soit le sous-problème est infaisable.

On rappelle qu'un problème d'optimisation sous contraintes est dit infaisable lorsqu'il n'existe pas de solution satisfaisant les contraintes du problème. Éliminer une branche signifie éliminer toutes les autres branches issues de cette dernière. Le sous-optimum *connu* est actualisé à chaque fois qu'une feuille est évaluée et que le coût obtenu est inférieur à celui du sous-optimum précédent.

Soit une séquence de modes \mathbf{I}_N (on fait l'hypothèse que le sous-problème QP associé cette séquence est faisable) où l'indice N signifie que la séquence est de longueur N . Soit un horizon P avec $P < N$. Les notations suivantes sont utilisées :

- $\mathbf{I}_P^{(N)}$ est la séquence des P premiers modes extraite de la séquence \mathbf{I}_N ;
- $\mathbf{U}_{cP}^{(N)}$ est la séquence de commandes continues de longueur P extraite de la séquence \mathbf{U}_{cN} .

On rappelle que l'exposant $*$ signifie l'optimalité vis à vis d'une séquence de modes donnée : *i.e.* la séquence \mathbf{U}_{cN}^* est optimale vis à vis d'une séquence de modes \mathbf{I}_N donnée. Toutefois, la séquence extraite $\mathbf{U}_{cP}^{*(N)}$ n'est pas nécessairement optimale sur l'horizon P .

Théorème : Soit une séquence de modes \mathbf{I}_N donnée, pour tout $P < N$, le coût optimal obtenu pour la séquence \mathbf{I}_N est supérieur au coût optimal obtenu pour une séquence

extraite $\mathbf{I}_P^{(N)}$:

$$\forall P < N, \quad J_N^*(\mathbf{x}(0), \mathbf{I}_N) \geq J_P^*(\mathbf{x}(0), \mathbf{I}_P^{(N)}) \quad (14)$$

Preuve : Le coût (12) se divise en deux termes, pour tout $P < N$:

$$J_N^*(\mathbf{x}(0), \mathbf{I}_N) = J_P(\mathbf{x}(0), (\mathbf{U}_{cP}^{*(N)}, \mathbf{I}_P^{(N)})) + \sum_{k=P}^{N-1} L(\mathbf{x}(k), (\mathbf{u}_c^*(k), i(k))) + F(\mathbf{x}(N)) \quad (15)$$

Le coût est additif et les fonctions L et F sont définies positives donc :

$$J_N^*(\mathbf{x}(0), \mathbf{I}_N) \geq J_P(\mathbf{x}(0), (\mathbf{U}_{cP}^{*(N)}, \mathbf{I}_P^{(N)})) \quad (16)$$

Le coût $J_P(\mathbf{x}(0), (\mathbf{U}_{cP}^{*(N)}, \mathbf{I}_P^{(N)}))$ est le coût calculé avec une séquence extraite $\mathbf{U}_{cP}^{*(N)}$ et est donc sous-optimal sur l'horizon P :

$$\underbrace{J_P(\mathbf{x}(0), (\mathbf{U}_{cP}^{*(N)}, \mathbf{I}_P^{(N)}))}_{\text{coût sous-optimal sur } P} \geq \underbrace{J_P^*(\mathbf{x}(0), \mathbf{I}_P^{(N)})}_{\text{coût optimal sur } P} \quad (17)$$

□

Supposons être à une profondeur $P < N$ et que le coût associé J_P est supérieur à un sous-optimum connu, en vertu du théorème énoncé ci-dessus, la branche correspondante et toutes les branches qui en sont issues peuvent être coupées.

Remarques :

1. La présence d'une contrainte égalité sur l'état final $\mathbf{x}(N) = \mathbf{x}_r$ (resp. un terme sur l'état final $F(\mathbf{x}(N))$) n'est à considérer que lors de l'évaluation d'un coût sur l'horizon complet (N), *i.e.* une feuille de l'arbre de la figure 1. Cette contrainte (resp. ce terme) n'existe pas lors de l'évaluation d'un coût partiel ($P < N$).

2. Dans le cas d'un problème d'optimisation purement combinatoire, le coût d'un nœud dans l'arbre des possibilités est obtenu par la somme des coûts associés aux branches menant à ce nœud (le coût est évalué en *descendant*). Le problème considéré ici est de nature mixte. Le coût d'une branche n'est pas connu a priori puisqu'il dépend de la commande continue \mathbf{u}_c^* trouvée en résolvant le sous-problème QP associé au chemin emprunté dans l'arbre. Soit un nœud à une profondeur $P < N$, le coût associé à ce nœud n'est pas la somme des coûts de branche mais le coût obtenu en résolvant un sous-problème QP sur l'horizon P .

L'algorithme d'énumération partielle est un algorithme de type B&B qui conduit à la solution optimale en exploitant la structure particulière du problème d'optimisation associé à la commande prédictive : la hauteur de l'arbre est fixée par la longueur de l'horizon de prédiction. Les techniques de programmation mixte entière à base de B&B n'exploitent pas cette caractéristique car elles utilisent un arbre de décision différent : si le problème comporte b variables entières supposées binaires, la profondeur de l'arbre est $N \times b$, il y a deux possibilités de branchement à chaque nœud et c'est le coût sur l'horizon complet qui est évalué. Elles ont été développées pour résoudre des problèmes standards, purement combinatoires ou mixtes. Dans le cadre de

la commande prédictive des systèmes MLD, le problème d'optimisation associé est reformulé en problème standard MIQP [5], [9].

La stratégie de descente proposée est une heuristique qui permet d'obtenir un premier sous-optimum qu'on espère de bonne qualité, *i.e.* peu éloigné de l'optimum. Le caractère sous-optimal provient du choix de la séquence de modes (stratégie du meilleur d'abord à un pas de prédiction) mais le sous-optimum est obtenu en résolvant un sous-problème QP sur l'horizon complet N .

Dans cette approche par énumération partielle, la dimension des sous-problèmes QP à résoudre commence à n_{uc} au début de l'arbre (prédiction à un pas) pour croître avec l'horizon jusqu'à $n_{uc} \times N$ en bas de l'arbre (horizon N).

V. EXEMPLE D'APPLICATION

A. Description du cas d'étude

L'algorithme proposé est appliqué au système à trois cuves [10] dont un schéma est représenté figure 2. Le sys-

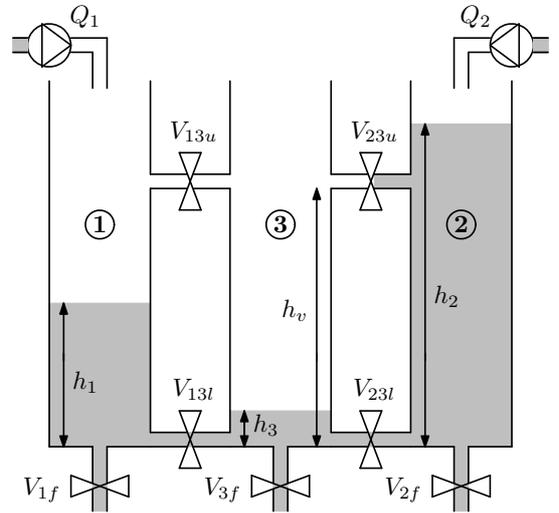


Fig. 2. Système à trois cuves

tème est composé de trois cuves numérotées de 1 à 3 de hauteur h_{max} . Les cuves 1 et 2 sont alimentées en eau par deux pompes dont les débits respectifs Q_1 et Q_2 peuvent varier entre 0 et Q_{max} . Quatre vannes V_{13l} , V_{23l} , V_{13u} et V_{23u} permettent de contrôler les flux entre les cuves. On suppose qu'elles ne peuvent prendre que deux états : ouvert (1) ou fermé (0). Les vannes supérieures sont situées à une hauteur h_v . Les vannes V_{1f} , V_{2f} et V_{3f} fixent les flux sortants et sont ouvertes à 10%. Les niveaux de liquide dans les trois cuves, qui sont les grandeurs à contrôler, sont notés h_1 , h_2 et h_3 respectivement.

La conservation de la masse dans les cuves permet d'écrire les équations différentielles suivantes :

$$\dot{h}_1 = \frac{1}{S_c} (Q_1 - Q_{13l} - Q_{13u} - Q_{1f}) \quad (18a)$$

$$\dot{h}_2 = \frac{1}{S_c} (Q_2 - Q_{23l} - Q_{23u} - Q_{2f}) \quad (18b)$$

$$\dot{h}_3 = \frac{1}{S_c} (Q_{13l} + Q_{13u} + Q_{23l} + Q_{23u} - Q_{3f}) \quad (18c)$$

où S_c est la section des cuves et les grandeurs Q sont les différents flux entre les cuves et flux sortants. Les débits

sont donnés par la loi de Toricelli et sont linéarisés en utilisant l'approximation suivante : $\sqrt{h} \approx h/\sqrt{h_{max}}$ pour $h \in [0, h_{max}]$ [9]. Les débits s'écrivent alors :

$$Q_{13l} \approx k_l V_{13l} (h_1 - h_3) \quad (19a)$$

$$Q_{23l} \approx k_l V_{23l} (h_2 - h_3) \quad (19b)$$

$$Q_{if} \approx k_l V_{if} h_i \text{ pour } i \in \{1, 2, 3\} \quad (19c)$$

avec $k_l = a S_v \sqrt{2g/h_{max}}$. a est une constante liée au liquide, S_v est la section des vannes et g est l'accélération de la pesanteur. Pour le débit Q_{13u} , il faut considérer les différents cas possibles pour les niveaux de liquide h_1 et h_3 :

- si $h_1 \geq h_v$ et $h_3 \geq h_v$ alors $Q_{13u} = k_u V_{13u} (h_1 - h_3)$;
- si $h_1 \leq h_v$ et $h_3 \geq h_v$ alors $Q_{13u} = k_u V_{13u} (h_v - h_3)$;
- si $h_1 \geq h_v$ et $h_3 \leq h_v$ alors $Q_{13u} = k_u V_{13u} (h_1 - h_v)$;
- si $h_1 \leq h_v$ et $h_3 \leq h_v$ alors $Q_{13u} = 0$.

avec $k_u = a S_v \sqrt{2g/(h_{max} - h_v)}$. Il en va de même pour le débit Q_{23u} dont l'expression, qui est analogue à Q_{13u} , n'est pas détaillée ici. Il résulte de ces considérations un partitionnement de l'espace d'état engendré par h_1 , h_2 et h_3 en huit régions contiguës. Les équations (18a)–(18c) sont discrétisées par la méthode d'Euler afin d'obtenir un modèle PWA à temps discret (1a)–(2).

Il y a huit partitions à considérer et les deux positions (ouverte ou fermée) des quatre vannes : le nombre total de modes est donc $8 \times 2^4 = 128$. La période d'échantillonnage est 10 s. On ne considère pas de coût ni de contrainte égalité sur l'état final. L'état initial est :

$$\mathbf{x} = (0.4 \ 0.9 \ 0.1)^T \quad (20)$$

Les consignes pour les niveaux de liquide sont :

$$\mathbf{y}_r = \mathbf{x}_r = (0.6 \ 0.8 \ 0.5)^T \quad (21)$$

Les pondérations utilisées dans la fonction de coût (6) sont :

$$\mathbf{Q}_u = \text{diag}(1 \ 1 \ 1 \ 1 \ 1) \quad (22)$$

$$\mathbf{Q}_y = \mathbf{Q}_x = \text{diag}(10^2 \ 10^2 \ 10^3) \quad (23)$$

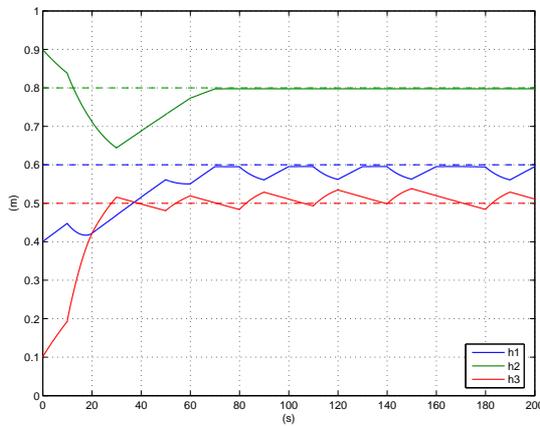


Fig. 3. Niveaux de liquide ($N = 3$)

Des simulations ont été effectuées pour différents horizons de prédiction ($2 \leq N \leq 6$). Les figures 3, 4 et 5 représentent respectivement les niveaux de liquide dans les

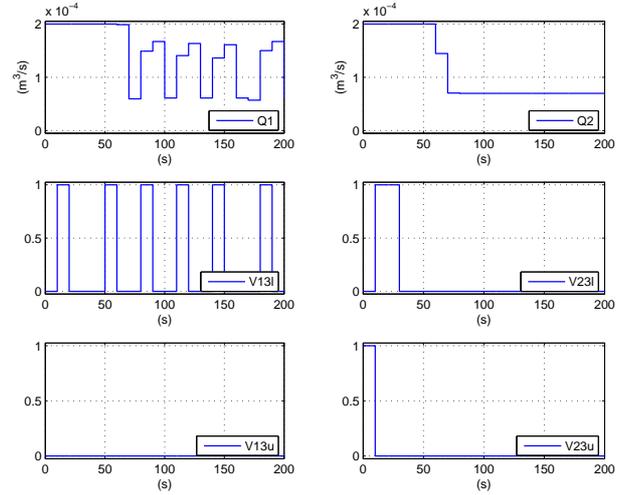


Fig. 4. Entrées de commande ($N = 3$)

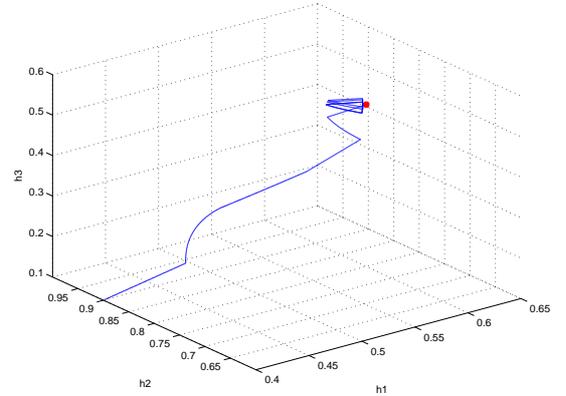


Fig. 5. Trajectoire dans l'espace d'état ($N = 3$)

trois cuves, les commandes continues et discrètes, et la trajectoire dans l'espace d'état obtenus pour $N = 3$. L'état à atteindre \mathbf{x}_r n'est pas un état d'équilibre pour le système et les niveaux des cuves 1 et 3 oscillent autour de leur valeur de consigne respective suivant l'ouverture et la fermeture de la vanne inférieure V_{13l} qui permet le passage du liquide entre ces deux cuves.

La figure 6 représente les nombres minimaux, moyens et maximaux de sous-problèmes QP résolus pour atteindre l'optimum recherché en fonction de la longueur N de l'horizon de prédiction. On peut observer que la complexité reste exponentielle mais que l'algorithme d'énumération partielle permet d'éviter l'explosion combinatoire.

Le tableau I met en lumière cet aspect pour les différents horizons employés. La première colonne donne les nombres de sous-problèmes QP qu'il faudrait résoudre à chaque instant d'échantillonnage en procédant par énumération exhaustive. Les trois colonnes suivantes donnent les nombres minimaux (QP min), moyens (QP moy) et maximaux (QP max) obtenus avec l'énumération partielle (le nombre de QP résolus à chaque instant varie sur la durée de la simulation). La dernière colonne donne les pourcentages correspondant aux nombres maximaux de QP résolus par rapport à l'énumération exhaustive. Pour $N = 3$ par ex., les optimisations effectuées lors de la simulation ont requis la résolution de 6528 QP au maximum à chaque instant, à comparer avec $128^3 \approx 2.1 \cdot 10^6$ (énumération exhaustive).

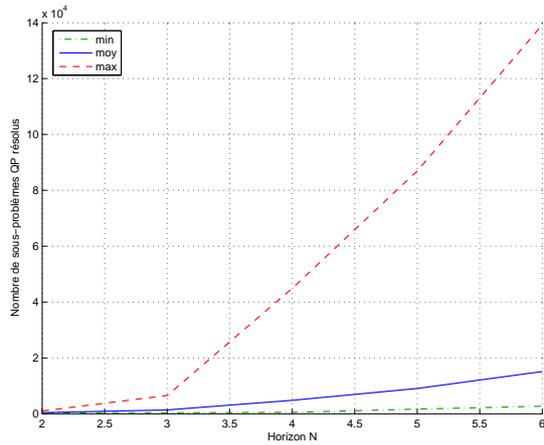


Fig. 6. Nombre de sous-problèmes résolus en fonction de l’horizon de prédiction

N	p^N	QP min	QP moy	QP max
2	16384	256	420	1024
3	$2.1 \cdot 10^6$	384	1390	6528
4	$2.7 \cdot 10^8$	512	4815	44928
5	$3.4 \cdot 10^{10}$	1664	9045	86912
6	$4.4 \cdot 10^{12}$	2688	15104	139264

TABLE I

PERFORMANCES DE L’ALGORITHME D’ÉNUMÉRATION PARTIELLE

Nous nous sommes intéressés à effectuer la comparaison des résultats obtenus avec les algorithmes d’énumération partielle et “miqp” [11] (associé au formalisme MLD, cité en introduction). La modélisation MLD du système à trois cuves a été obtenue à l’aide du logiciel Hysdel [12]. Les résultats sont présentés dans le tableau II pour $2 \leq N \leq 4$ avec les nombres maximaux de sous-problèmes résolus et les temps de calcul associés. Ces résultats ont été obtenus avec un ordinateur de type PC équipé d’un processeur Centrino cadencé à 2 GHz et de la version 7.0 du logiciel Matlab. Les algorithmes utilisés ont été développés en code Matlab et le solveur utilisé pour résoudre les sous-problèmes QP est “quadprog” issu de la boîte à outils “Optimization”.

Outre le nombre de sous-problèmes résolus pour obtenir l’optimum, le temps de calcul nécessaire est aussi lié à la dimension des sous-problèmes. Le formalisme MLD requiert l’ajout de $n_\delta = 3$ variables auxiliaires binaires et $n_z = 7$ variables auxiliaires réelles. La dimension maximale d’un sous-problème est alors $(n_{uc} + n_{ud} + n_\delta + n_z) \times N$. Pour l’énumération partielle, la dimension maximale d’un sous-problème est $n_{uc} \times N$. L’intérêt de l’approche proposée réside aussi dans la faible dimension des sous-problèmes à résoudre, ce qui induit de faibles temps de calcul. Au début de l’optimisation, il faut résoudre un nombre important de sous-problèmes mais de faibles dimensions. Le fait de couper des branches dans l’arbre des possibilités restreint le nombre de sous-problèmes de taille plus importante à résoudre.

VI. CONCLUSIONS

Cet article présente un algorithme simple et efficace pour résoudre le problème d’optimisation mixte associé à la com-

N	énum. part.		miqp (MLD)	
	QP max	temps (s)	QP max	temps (s)
2	1024	4.34	589	35.1
3	6528	27.8	6821	591
4	44928	233.3	27609	3583

TABLE II

PERFORMANCES COMPARÉES

mande prédictive des systèmes PWA avec entrées mixtes. Connaissant un sous-optimum, l’idée clé de cet algorithme est d’évaluer des coûts partiels (*i.e.* à des horizons intermédiaires) afin d’élaguer l’arbre des possibilités en coupant les branches qui ne peuvent pas conduire à l’optimum recherché.

Les résultats de simulation obtenus avec le cas d’étude du système à trois cuves permettent de juger de l’efficacité de l’approche proposée et d’effectuer une comparaison avec l’algorithme “miqp” associé au formalisme MLD.

Des travaux futurs consisteront à programmer l’algorithme proposé dans un langage rapide. Il sera alors possible d’effectuer des comparaisons avec des solveurs commerciaux puissants (et coûteux) tels que CPLEX qui sont incontournables pour l’utilisation du formalisme MLD avec des problèmes non triviaux.

RÉFÉRENCES

- [1] M. Peña, E. F. Camacho, et S. Piñón. Hybrid systems for solving model predictive control of piecewise affine system. *Proceedings of the IFAC Conference on Analysis and Design of Hybrid Systems*, Saint-Malo, France, 2003.
- [2] A. Bemporad, F. Borrelli, et M. Morari. Model Predictive Control Based on Linear Programming - The Explicit Solution. *IEEE Transactions on Automatic Control*, 47(12):1974–1985, December 2002.
- [3] S. Leirens, J. Buisson, P. Bastard, et J.-L. Coullon. An efficient algorithm for solving model predictive control of switched affine systems. *Proceedings of the Scientific Computation, Applied Mathematics and Simulation World Congress*, Paris, France, 2005.
- [4] W. P. M. H. Heemels, B. De Schutter, et A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, 2001.
- [5] A. Bemporad et M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.
- [6] J. Thomas, J. Buisson, D. Dumur, et H. Guéguen. Model predictive control for hybrid systems under a state partition based mld approach (SPMLD). *Proceedings of the International Conference on Informatics in Control, Automation and Robotics*, Setúbal, Portugal, 2004.
- [7] A. Bemporad et N. Giorgetti. A SAT-based hybrid solver for optimal control of hybrid systems. R. Alur et G.J. Pappas, editors, *Hybrid Systems: Computation and Control*, number 2993 in Lecture Notes in Computer Science, pages 126–141, 2004.
- [8] D. Q. Mayne, J. B. Rawlings, C. V. Rao, et P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.
- [9] D. Mignone. *Control and Estimation of Hybrid Systems with Mathematical Optimization*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zürich, 2002.
- [10] J. Lunze. Laboratory three tanks system benchmark for the re-configuration problem. Technical report, Technical University of Hamburg-Harburg, Institute of Control Engineering, Hamburg, Germany, 1998.
- [11] A. Bemporad et D. Mignone. miqp.m : a Matlab function for solving mixed integer quadratic programs. Technical report, Automatic Control Laboratory, ETH, Zürich, Switzerland, 2000.
- [12] F. Torrisi, A. Bemporad, et D. Mignone. Hysdel - a tool for generating hybrid models. Technical Report AUT00-03, Automatic Control Laboratory, ETH, Zürich, Switzerland, 2002.