



HAL
open science

Analyse Algorithmique des Systèmes Hybrides

Antoine Girard

► **To cite this version:**

Antoine Girard. Analyse Algorithmique des Systèmes Hybrides. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG, 2004. Français. NNT: . tel-00007064

HAL Id: tel-00007064

<https://theses.hal.science/tel-00007064v1>

Submitted on 8 Oct 2004

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.

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

THESE

pour obtenir le grade de

DOCTEUR DE L'INPG

Spécialité : Mathématiques Appliquées

préparée au Laboratoire de Modélisation et Calcul (LMC / IMAG)
dans le cadre de l'**Ecole Doctorale**
Mathématiques, Sciences et Technologies de l'Information

présentée et soutenue publiquement par

Antoine GIRARD

le 30 septembre 2004

**Analyse Algorithmique des
Systèmes Hybrides**

Directeur de thèse :

Jean DELLA DORA

JURY

M.	Oded MALER	, Président
M.	Jacques BLUM	, Rapporteur
M.	Jean-Pierre FRANCOISE	, Rapporteur
M.	Eugene ASARIN	, Examineur
M.	Jean DELLA DORA	, Examineur
Mme	Valérie PERRIER	, Examinatrice

Remerciements

Je tiens tout d'abord à remercier mon directeur de thèse, Jean Della Dora pour son enthousiasme, son aide et la confiance qu'il m'a accordée tant sur le plan de la recherche que de l'enseignement.

Je remercie Messieurs Jacques Blum et Jean-Pierre Françoise pour avoir accepté d'être les rapporteurs de cette thèse. L'attention et l'intérêt qu'ils ont portés à mon travail m'honorent.

Ma reconnaissance va ensuite à M. Oded Maler, président du jury, pour l'intérêt qu'il a montré pour mon travail durant ces trois années et pour l'aide précieuse qu'il m'a apportée dans la préparation de l'après thèse.

Enfin, je remercie Valérie Perrier et Eugene Asarin pour avoir accepté d'être membres de mon jury. En particulier, je souhaite exprimer ma gratitude à Eugene Asarin ainsi qu'à Thao Dang pour m'avoir fait l'honneur de me proposer une collaboration. Travailler avec eux a été pour moi une expérience aussi instructive qu'agréable.

Ce travail a été financé par la région Rhône-Alpes. Je les remercie de la confiance qu'ils m'ont accordée et qu'ils ont renouvelée en m'attribuant une bourse internationale de recherche pour l'année 2004-2005.

Mon amitié va à mes collègues du bureau 14, Etienne, Laurent et Olivier : merci pour tous ces moments passés au labo ou en dehors à parler de maths (un peu) et d'autre chose (beaucoup).

Mes remerciements vont également aux membres de l'équipe Mosaic et aux thésards du Laboratoire de Modélisation et Calcul (ou d'ailleurs). En particulier, je tiens à saluer Françoise, Jean-Guillaume, Julie, Laurent, Aude, Pierre, Alex, Aude, Claire, Basile, Anne, Sylvain, Vincent, François, Guillaume, Erwan, Clément, Morgan, Alex, Sophie, Jocelyn...

Mes remerciements vont à ma famille, à mes parents, à Adrien et à Anne-Laure. Je tiens particulièrement à remercier mes parents pour leur soutien et leur aide.

Enfin, et surtout, merci à Anne...

Table des matières

I	Introduction à la Théorie des Systèmes Hybrides	13
1	Définition et Exemples	15
1.1	Définition d'un système hybride	15
1.2	Exemples de systèmes hybrides	17
1.2.1	Systèmes dynamiques impulsionnels	17
1.2.2	Systèmes dynamiques par morceaux	18
1.2.3	Systèmes dynamiques à commutation	20
1.2.4	Des systèmes à commutation aux systèmes hybrides	21
2	Notion de Solution d'un Système Hybride	23
2.1	Exécution d'un système hybride	23
2.1.1	Trajectoire temporisée	23
2.1.2	Trajectoire hybride	24
2.1.3	Définition d'une exécution d'un système hybride	26
2.1.4	Classification des exécutions	26
2.2	Systèmes hybrides déterministes et non-bloquants	29
2.2.1	Caractérisation d'un système hybride déterministe	30
2.2.2	Caractérisation d'un système hybride non-bloquant	30
2.2.3	Existence et unicité des exécutions	31
2.3	Exécution Zénon	31
3	Systèmes Hybrides Linéaires par Morceaux	35
3.1	Définition du modèle hybride linéaire par morceaux	35
3.2	Discussion sur l'existence et l'unicité des exécutions	37
3.2.1	Systèmes autonomes	37
3.2.2	Systèmes avec entrée	39
II	Calcul des Exécutions d'un Système Hybride	41
4	Simulation des Systèmes Hybrides	43
4.1	Simulation de l'évolution continue	45
4.2	Détection des événements	45
4.2.1	Supprimer les contraintes superflues	47
4.2.2	Algorithme de résolution	48
4.2.3	Convergence et tests	52

5	Exécutions Périodiques d'un Système Hybride	55
5.1	Définition, exemple	56
5.2	Application de Poincaré hybride	58
5.2.1	Application de Poincaré pour les systèmes dynamiques	59
5.2.2	Application d'impact	59
5.2.3	Application de Poincaré hybride	62
5.3	Stabilité et calcul des exécutions périodiques	63
5.3.1	Stabilité du cycle hybride	63
5.3.2	Calcul du cycle hybride	64
5.4	Exemple	65
 III Analyse d'Atteignabilité		 71
6	Atteignabilité des Systèmes Dynamiques Linéaires	73
6.1	Approximation d'ensembles	74
6.2	Atteignabilité des systèmes linéaires autonomes	75
6.2.1	Calcul de l'image par le flot	76
6.2.2	Approximation conservative de $\mathcal{R}_{[0,r]}(I)$	77
6.2.3	L'approche de d/dt	80
6.2.4	L'approche de CheckMate	82
6.2.5	Comparaison des deux méthodes	84
6.3	Atteignabilité des systèmes linéaires soumis à des perturbations	85
6.3.1	Approximation conservative de l'image par le flot	86
6.3.2	Approximation conservative de $\hat{\mathcal{R}}_{[0,r]}^\mu(I)$	88
6.3.3	Approximation conservative de l'ensemble atteignable	90
6.3.4	Exemple	94
7	Atteignabilité des Systèmes Hybrides	97
7.1	Ensemble atteignable d'un système hybride	97
7.2	Calcul des successeurs continus	99
7.3	Calcul de l'ensemble atteignable d'un système hybride	101
7.4	Exemple	105
 IV Contrôle des Systèmes Hybrides		 109
8	Analyse Multirésolution et Théorie du Contrôle	111
8.1	Quelques résultats de la théorie du contrôle	113
8.1.1	Ensemble des entrées optimales : définition, caractérisation	113
8.1.2	Contrôlabilité	116
8.2	Une analyse multirésolution de $\mathcal{O}_{(A,B)}$	117
8.2.1	Analyse multirésolution et base d'ondelettes	117
8.2.2	Analyse multirésolution de la théorie du contrôle	118
8.2.3	Ondelettes de la théorie du contrôle	119
8.2.4	Base des fonctions sorties	123

9	Contrôle Optimal des Systèmes Hybrides	127
9.1	Discrétisation du problème de contrôle optimal	128
9.2	Régularisation de la sortie	130
9.3	Exemples	133
9.3.1	Résultats expérimentaux	133
9.3.2	Contrôle optimal d'un système hybride	135
V	Approximation de Dynamiques Non-Linéaires	139
10	Calcul Hybride : Equation Différentielle Scalaire	141
10.1	Exemple introductif	141
10.2	Approximation des solutions d'équations différentielles	143
10.2.1	Méthode générale	143
10.2.2	Convergence de l'approximation	144
10.2.3	Résultats expérimentaux	145
10.2.4	Limites de la méthode	146
10.3	Améliorations de la convergence	147
10.3.1	Interpolation aux abscisses de Gauss	147
10.3.2	Interpolation quadratique par morceaux	149
10.4	Résultats expérimentaux	150
11	Calcul Hybride en Dimension Supérieure	153
11.1	Approximation de l'équation différentielle	153
11.1.1	Approximation linéaire par morceaux d'un champ de vecteurs	154
11.1.2	Maillage simplicial du domaine	159
11.2	Approximation des solutions de l'équation différentielle	161
11.2.1	Approximation de la solution du problème de Cauchy	161
11.2.2	Approximation des attracteurs	162
11.2.3	Comparaison des portraits de phases	164
11.3	Approximation conservative	165

Introduction

Historiquement, l'étude des systèmes a été abordée du point de vue d'une modélisation continue au moyen d'équations différentielles.

Les progrès de l'informatique, et l'utilisation d'ordinateurs dans des processus de commande, ont motivé le développement d'outils théoriques permettant une description plus qualitative de la dynamique des systèmes, comme les automates finis.

Cependant, dans de nombreux cas, une telle description échoue à rendre la complexité et la richesse des comportements des systèmes. Ainsi, la connaissance d'une modélisation continue de la dynamique est souvent nécessaire à la synthèse d'une loi de commande implémentable par un automate. L'objet mathématique résultant du couplage des équations différentielles modélisant l'évolution du système et de l'automate implémentant la loi de commande forme ce que l'on appelle un automate hybride.

L'importance du problème a entraîné un développement rapide de la théorie des systèmes hybrides. D'une manière générale, un système hybride est un système dont l'évolution au cours du temps est décrite par un ensemble de lois mathématiques qui peuvent être de natures continues ou discrètes. Hormis les problèmes de commande par ordinateur, cette théorie a trouvé de nombreuses applications dans des domaines aussi variés que l'industrie automobile, la robotique, le contrôle du trafic aérien, la biologie...

La classe des systèmes hybrides linéaires par morceaux est probablement l'une des plus importantes. Un système hybride est dit linéaire par morceaux si les lois décrivant son évolution continue sont formulées au moyen d'équations différentielles linéaires. Un effort particulier a été apporté à l'étude de cette classe pour deux raisons principales. D'abord, elle est suffisamment riche pour permettre une modélisation réaliste de nombreux problèmes. Ensuite, sa simplicité relative permet la conception d'outils algorithmiques pour l'analyse de ces systèmes.

Cette thèse est consacrée à l'analyse algorithmique des systèmes hybrides linéaires par morceaux, elle est composée de cinq parties :

Introduction à la Théorie des Systèmes Hybrides

Dans cette première partie, nous introduisons les notions fondamentales de la théorie.

Le premier chapitre est consacré à la définition d'un système hybride. La définition est illustrée par de nombreux exemples de modélisation venant de domaines variés comme la mécanique, l'électronique où l'ingénierie automobile.

Le deuxième chapitre formalise la notion d'exécution d'un système hybride. Nous procédons à une classification des exécutions (infinies, maximales, Zénon) et des systèmes hybrides (déterministes, non-bloquants). Nous énonçons un théorème d'existence et d'unicité d'une exécution infinie acceptée par le système hybride, pour n'importe quelle condition initiale.

Enfin, le troisième chapitre présente les restrictions associées à la classe des systèmes hybrides linéaires par morceaux à laquelle notre étude est consacrée.

Calcul des Exécutions d'un Système Hybride

Cette partie est dédiée au calcul algorithmique des exécutions acceptées par un système hybride où la dynamique continue est donnée par des équations différentielles linéaires autonomes.

Le quatrième chapitre est consacré au problème de la simulation des systèmes hybrides linéaires par morceaux et notamment au processus de détection des événements.

Dans le cinquième chapitre, nous nous intéressons aux exécutions périodiques. Nous généralisons le concept d'application de Poincaré aux systèmes hybrides. Cet outil nous permet de proposer un algorithme de calcul des exécutions périodiques ainsi qu'un critère permettant d'évaluer leur stabilité. Nous terminons ce chapitre en réalisant l'étude d'un système composé de deux réservoirs exhibant un comportement asymptotique périodique.

Analyse d'Atteignabilité

Dans cette partie, nous proposons un algorithme de calcul de l'ensemble atteignable d'un système hybride. Nous apportons un soin particulier aux systèmes où les dynamiques continues sont connues de manière incertaine.

Dans le sixième chapitre, nous présentons dans un premier temps les deux approches existantes pour le calcul d'une approximation de l'ensemble atteignable des systèmes dynamiques linéaires autonomes. Nous proposons ensuite une généralisation d'une de ces deux approches aux systèmes dynamiques linéaires soumis à des perturbations.

Dans le septième chapitre, nous montrons comment la méthode développée dans le chapitre précédent peut être incorporée dans un algorithme de calcul de l'ensemble atteignable d'un système hybride linéaire par morceaux. Nous complétons ensuite l'étude du système de deux réservoirs commencée au chapitre précédent en supposant que la dynamique du système est soumise à des fluctuations incontrôlables.

Contrôle des Systèmes Hybrides

Dans cette partie, nous proposons une méthodologie pour le contrôle des systèmes hybrides linéaires par morceaux.

Le huitième chapitre est consacré au développement d'une analyse multirésolution de l'espace des entrées d'un système dynamique linéaire. Il en résulte une base d'ondelettes de la théorie du contrôle permettant une analyse hiérarchique et locale des signaux d'entrée d'un système linéaire.

Dans le neuvième chapitre, nous montrons comment les bases d'ondelettes de la théorie du contrôle peuvent être utilisées pour la synthèse d'une solution approchée d'un problème

de contrôle optimal d'un système hybride linéaire par morceaux. L'avantage de cette méthode est que l'exécution associée à l'entrée calculée est effectivement acceptée par le système hybride.

Approximation de Dynamiques Non-Linéaires

Cette dernière partie est consacrée à l'approximation de solutions d'équations différentielles non-linéaires par des méthodes de calcul hybride.

Dans le dixième chapitre, nous montrons comment l'objet *système hybride* peut être utilisé comme outil d'approximation de dynamiques non-linéaires. Dans ce chapitre, nous considérons plus particulièrement des équations différentielles scalaires.

Dans le onzième et dernier chapitre, nous généralisons l'approche développée dans le chapitre précédent aux équations différentielles en dimension supérieure.

Première partie

Introduction à la Théorie des
Systèmes Hybrides

Chapitre 1

Définition et Exemples

1.1 Définition d'un système hybride

Tout système impliquant des processus continus et des phénomènes discrets peut être vu comme un système hybride. Par extension, lorsque dans un même système physique certaines grandeurs varient très rapidement (quasi-instantanément) par rapport aux autres, alors, une modélisation hybride de ce système est envisageable et donne souvent de bien meilleurs résultats qu'une modélisation continue.

Ainsi, les domaines d'application des systèmes hybrides sont extrêmement nombreux et variés; on peut citer entre autres l'informatique [73], l'industrie automobile [16], la robotique [10], le contrôle du trafic aérien [93], la biologie [12]...

Par conséquent, une définition unifiée des systèmes hybrides pouvant servir d'environnement théorique à la description de ces phénomènes est difficile. La définition suivante nous semble relativement générale. Elle est notamment motivée par [17, 87].

Définition 1.1.1 (Système hybride) *Un système hybride est un septuple*

$$\mathcal{H} = (\mathcal{Q}, \mathcal{E}, \mathcal{D}, \mathcal{U}, \mathcal{F}, \mathcal{G}, \mathcal{R})$$

où :

1. \mathcal{Q} est l'ensemble dénombrable des états discrets (ou modes).
2. $\mathcal{E} \subseteq \mathcal{Q} \times \mathcal{Q}$ est l'ensemble des arêtes (ou transitions).
3. $\mathcal{D} = \{D_q, q \in \mathcal{Q}\}$ est la collection des domaines.
 $\forall q \in \mathcal{Q}, D_q$ est un sous-ensemble de \mathbb{R}^n d'intérieur non-vide.
4. $\mathcal{U} = \{U_q, q \in \mathcal{Q}\}$ est la collection des domaines de contrôle.
 $\forall q \in \mathcal{Q}, U_q$ est un sous-ensemble de \mathbb{R}^p .
5. $\mathcal{F} = \{f_q, q \in \mathcal{Q}\}$ est la collection des champs de vecteurs.
 $\forall q \in \mathcal{Q}, f_q : D_q \times U_q \rightarrow \mathbb{R}^n$.
6. $\mathcal{G} = \{G_e, e \in \mathcal{E}\}$ est la collection des gardes.
 $\forall e = (q, q') \in \mathcal{E}, G_e \subseteq D_q$.
7. $\mathcal{R} = \{R_e, e \in \mathcal{E}\}$ est la collection des fonctions resets.
 $\forall e = (q, q') \in \mathcal{E}, R_e : G_e \rightarrow 2^{D_{q'}}$ où $2^{D_{q'}}$ dénote l'ensemble des parties de $D_{q'}$.
On suppose que pour tout $x \in G_e, R_e(x) \neq \emptyset$.

Remarque 1.1.1 *Il arrive que les équations différentielles associées aux éléments de \mathcal{F} soient autonomes (i.e. $f_q : D_q \rightarrow \mathbb{R}^n$). Dans ce cas, la donnée des domaines de contrôle est superflue et le système hybride est alors défini par le sextuple $(\mathcal{Q}, \mathcal{E}, \mathcal{D}, \mathcal{F}, \mathcal{G}, \mathcal{R})$.*

Intuitivement, l'état du système hybride peut être décrit par deux variables : la première discrète, notée $q(t)$ à valeur dans \mathcal{Q} , la deuxième continue, notée $x(t)$ à valeur dans \mathbb{R}^n .

Un exemple de fonctionnement (on parle d'exécution) est montré sur la figure 1.1. A l'instant initial ($t = t_0$), $q(t_0) = q_0$ et la valeur initiale de la variable continue $x(t_0)$ est un élément du domaine D_{q_0} . Cette dernière évolue alors en suivant l'équation différentielle $x'(t) = f_{q_0}(x(t), u(t))$, où la fonction u prend ses valeurs dans le domaine de contrôle U_{q_0} . Enfin, lorsque la trajectoire $x(t)$ atteint une garde G_e à l'instant t_1 (avec ici $e = (q_0, q_1)$), la variable discrète $q(t)$ peut alors prendre la valeur q_1 . La variable continue est alors réinitialisée à une valeur de l'ensemble $R_e(x(t_1^-)) \subseteq D_{q_1}$. On répète alors le même processus avec une nouvelle équation différentielle et de nouvelles gardes.

Par la suite, nous définirons rigoureusement la notion d'exécution d'un système hybride.

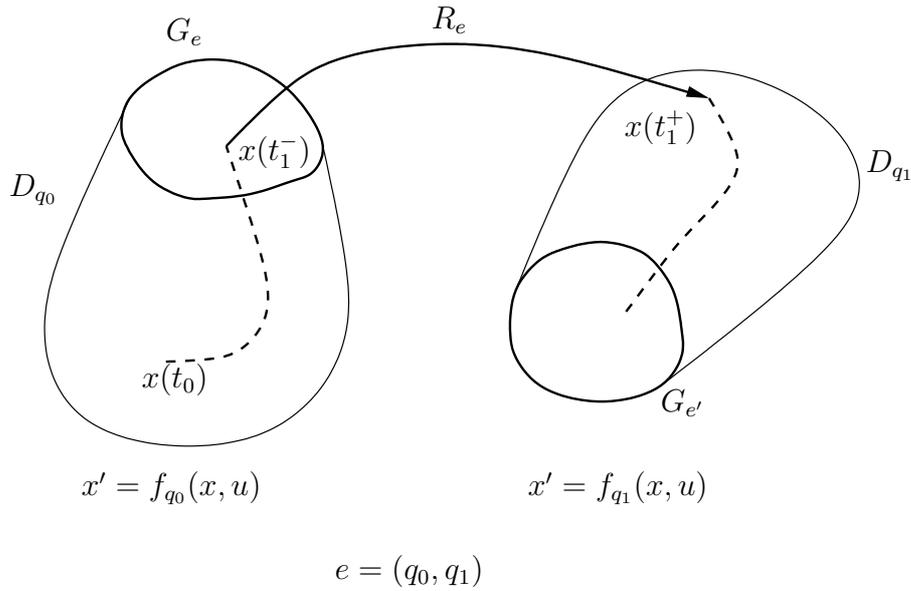


FIG. 1.1: Exemple d'exécution d'un système hybride

Un système hybride est donc constitué de deux composantes complémentaires couplées.

La première, donnée par le couple $(\mathcal{Q}, \mathcal{E})$, détermine la dynamique discrète du système. Elle est décrite par l'automate dont les sommets sont les éléments de \mathcal{Q} et les arêtes ceux de \mathcal{E} .

La deuxième composante du système hybride est définie par le triplet $(\mathcal{D}, \mathcal{U}, \mathcal{F})$ et décrit la dynamique continue du système. Elle consiste en une collection de systèmes dynamiques indexés sur les éléments de \mathcal{Q} . Ainsi, la valeur de la variable discrète $q(t)$ détermine l'équation différentielle gouvernant l'évolution de la variable continue $x(t)$. A

chaque instant t , tel que $q(t) = q$,

$$\begin{cases} x'(t) = f_q(x(t), u(t)), \\ (x(t), u(t)) \in D_q \times U_q. \end{cases}$$

Le couplage des deux composantes se fait par l'introduction de l'ensemble des gardes \mathcal{G} . En effet, pour qu'une transition de l'automate $e \in \mathcal{E}$ puisse se faire à un instant t , il est nécessaire que $x(t)$ appartienne à l'ensemble G_e .

Ainsi, la composante discrète du système contrôle la composante continue qui, rétro-activement, détermine, à chaque instant, l'ensemble (qui peut être vide) des transitions possibles de l'automate. L'ensemble résultant constitue ce que l'on appelle l'automate hybride.

1.2 Exemples de systèmes hybrides

Le formalisme des systèmes hybrides est très général et englobe de nombreuses classes de modèles de la théorie du contrôle. Dans ce paragraphe, nous présentons certaines d'entre elles qui nous semblent significatives.

1.2.1 Systèmes dynamiques impulsionsnels

Un système dynamique impulsionsnel [82] décrit l'évolution d'une variable continue $x(t)$ régie par une équation différentielle sous contraintes. Lorsque $x(t)$ vérifie certaines conditions, il est alors possible de lui donner une *impulsion*, c'est à dire de lui affecter une nouvelle valeur.

Dans le cadre des systèmes hybrides, les systèmes dynamiques impulsionsnels correspondent aux systèmes possédant un seul mode ($\mathcal{Q} = \{q\}$) et une transition $\mathcal{E} = \{e = (q, q)\}$ autorisant la réinitialisation de la variable continue via la fonction reset R_e .

Exemple 1.2.1 (Balle bondissante [64]) *On considère une balle de masse m soumise à l'action de la gravité. On la laisse tomber 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 $mz''(t) = -mg$. Quand $z(t) = 0$, la balle touche le sol et rebondit en perdant une fraction de son énergie :*

$$z'(t^+) = -cz'(t^-), \text{ avec } c \leq 1.$$

En posant $x_1(t) = z(t)$, $x_2(t) = z'(t)$ et en utilisant le formalisme de la définition 1.1.1, le modèle hybride de la balle bondissante est donné par :

1. $\mathcal{Q} = \{q\}$
2. $\mathcal{E} = \{e = (q, q)\}$
3. $D_q = \mathbb{R}^+ \times \mathbb{R}$
4. $f_q(x_1, x_2) = (x_2, -g)$
5. $G_e = \{x_1 = 0\}$
6. $R_e(x_1, x_2) = \{(x_1, -cx_2)\}$

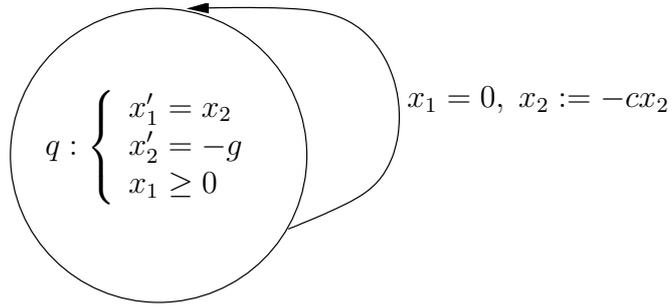


FIG. 1.2: Automate hybride de la balle bondissante

1.2.2 Systèmes dynamiques par morceaux

Les équations différentielles à second membre discontinu [41] ou défini par morceaux interviennent dans de nombreux problèmes d'ingénierie; notamment en électronique où les modèles de composants sont souvent linéaires par morceaux [19].

Le domaine de définition D de l'équation différentielle est un sous-ensemble fermé et connexe de \mathbb{R}^n . D est découpé en sous-domaines $\{D_q, q \in \mathcal{Q}\}$, fermés, d'intérieur non-vide et deux à deux disjoints, et tels que :

$$\bigcup_{q \in \mathcal{Q}} D_q = D.$$

Sur chaque domaine D_q , on définit un champ de vecteurs f_q .

La trajectoire $x(t)$ du système dynamique par morceaux se construit de la manière suivante (voir figure 1.3). Si $x(t_0)$ appartient à l'intérieur du domaine D_{q_0} , alors $x(t)$ est solution de l'équation différentielle associée au champ de vecteurs f_{q_0} jusqu'à l'instant t_1 où $x(t)$ atteint la frontière séparant le domaine D_{q_0} du domaine D_{q_1} . $x(t)$ devient alors solution de l'équation différentielle associée au champ de vecteurs f_{q_1} .

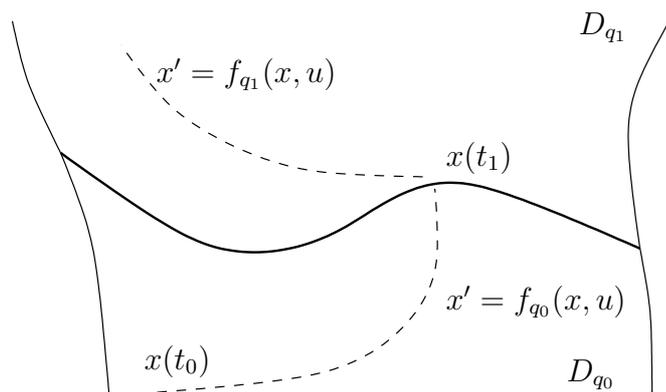


FIG. 1.3: Trajectoire du système dynamique par morceaux

Dans le cadre de la définition 1.1.1, un système dynamique par morceaux est donc un système hybride présentant les particularités suivantes.

- \mathcal{Q} est l'ensemble indexant les sous domaines D_q .

- $\mathcal{E} = \{(q, q') \in \mathcal{Q} \times \mathcal{Q}, \partial D_q \cap \partial D_{q'} \neq \emptyset\}$.
On ne peut passer du domaine D_q au domaine $D_{q'}$ que s'ils ont une frontière commune.
- $\mathcal{G} = \{G_e, e \in \mathcal{E}\}$. $\forall e = (q, q') \in \mathcal{E}, G_e = \partial D_q \cap \partial D_{q'}$.
On ne peut passer du domaine D_q au domaine $D_{q'}$ qu'en franchissant leur frontière commune.
- $\mathcal{R} = \{R_e, e \in \mathcal{E}\}$. $\forall e \in \mathcal{E}, \forall x \in G_e, R_e(x) = \{x\}$.
Il n'y a pas de réinitialisation de la variable continue.

Exemple 1.2.2 (Oscillateur électronique à valve [1, 79]) *L'oscillateur électronique à valve est un circuit composé d'une valve électronique, d'un circuit oscillant de type RLC et d'un dispositif de rétroaction par induction électromagnétique (voir figure 1.4). Une attention particulière a été portée à ce montage car c'est le circuit le plus simple exhibant des oscillations de type cycle limite.*

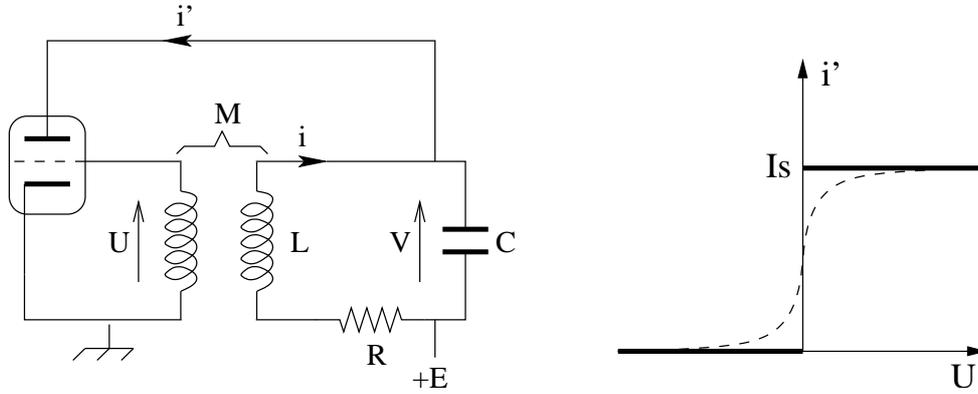


FIG. 1.4: Gauche : oscillateur électronique à valve. Droite : caractéristique de la valve.

En appliquant la loi des mailles dans le circuit RLC et la loi des noeuds au point de branchement de la valve électronique et du circuit RLC, on obtient les équations

$$Ri + V + L \frac{di}{dt} = 0, \quad i = i' + C \frac{dV}{dt} \implies LC \frac{d^2i}{dt^2} + RC \frac{di}{dt} + i = i'.$$

Le courant entrant à l'anode de la valve est donné par l'intensité i' et dépend exclusivement de la tension U appliquée à la grille. La caractéristique de la valve $i' = i'(U)$ est représentée en pointillé sur la figure 1.4.

Si les oscillations de la tension U sont d'amplitude suffisante, l'intensité i' est, la plupart du temps, égale soit à 0 soit à I_s . On peut alors, raisonnablement, représenter les propriétés d'une telle valve par la caractéristique idéale dessinée en gras sur la figure. Par conséquent,

$$LC \frac{d^2i}{dt^2} + RC \frac{di}{dt} + i = \begin{cases} 0 & \text{si } U < 0 \\ I_s & \text{si } U > 0. \end{cases}$$

La tension U est la différence de potentiels aux bornes de la bobine couplée à la bobine du circuit RLC. Le courant d'intensité i crée un champ magnétique à l'intérieur de la bobine du circuit RLC. Ce champ magnétique, à son tour, induit un courant dans la bobine reliée

à la grille de la valve. Cette action rétroactive se traduit par l'équation

$$U = -M \frac{di}{dt}.$$

On suppose que les bobines sont agencées de manière à avoir $M < 0$. On a donc

$$LC \frac{d^2i}{dt^2} + RC \frac{di}{dt} + i = \begin{cases} 0 & \text{si } \frac{di}{dt} < 0 \\ I_s & \text{si } \frac{di}{dt} > 0. \end{cases}$$

En posant, $x_1 = i$, $x_2 = \frac{di}{dt}$, l'oscillateur électronique à valve peut être modélisé par le système dynamique par morceaux :

$$x'_1(t) = x_2(t), \quad x'_2(t) = \begin{cases} -\frac{1}{LC}x_1(t) - \frac{R}{L}x_2(t) & \text{si } x_2(t) < 0 \\ -\frac{1}{LC}x_1(t) - \frac{R}{L}x_2(t) + \frac{I_s}{LC} & \text{si } x_2(t) > 0. \end{cases}$$

L'automate hybride correspondant est représenté sur la figure 1.5.

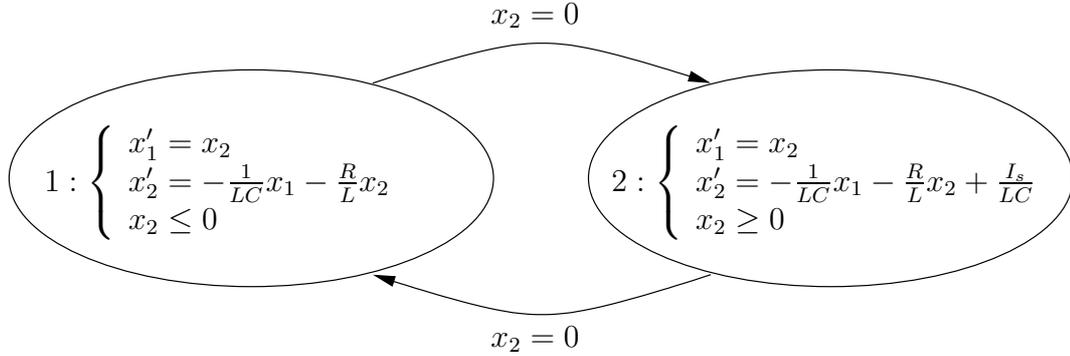


FIG. 1.5: Automate hybride de l'oscillateur électrique à valve

1.2.3 Systèmes dynamiques à commutation

Un système dynamique à commutation ou *switched system* (voir par exemple [37]) est un système hybride où la variable discrète $q(t)$ n'est pas vue comme une variable d'état mais comme une variable de contrôle. Ainsi, l'évolution de $q(t)$ n'est pas contrainte par un système de gardes mais donnée par un individu extérieur.

Par conséquent, d'après la définition 1.1.1, les systèmes dynamiques à commutation vérifient la propriété suivante :

$$\forall e = (q, q') \in \mathcal{E}, \quad G_e = D_{q'}.$$

Exemple 1.2.3 (Boîte de vitesse [43]) Nous présentons ici un modèle simplifié d'un véhicule motorisé à deux vitesses se déplaçant sur un axe. On note $x_1(t)$ la position du véhicule sur l'axe et $x_2(t)$ sa vitesse.

Le conducteur agit sur le système par deux moyens. D'abord, via la pédale d'accélération, il contrôle l'apport de carburant représenté par la variable $u(t) \in [0, u_{max}]$. Ensuite, le véhicule est équipé d'une boîte de vitesse à deux rapports, on note $q(t) \in \{1, 2\}$ le rapport actif.

Le système est décrit par les équations

$$x_1'(t) = x_2(t), \quad x_2'(t) = u(t)g_{q(t)}(x_2(t)) - kx_2(t).$$

Les deux rapports de la boîte de vitesse sont caractérisés par les fonctions g_1 et g_2 (figure 1.6). Elles représentent le rendement du moteur et relie l'accélération du véhicule, sa vitesse, l'apport de carburant et le rapport actif.

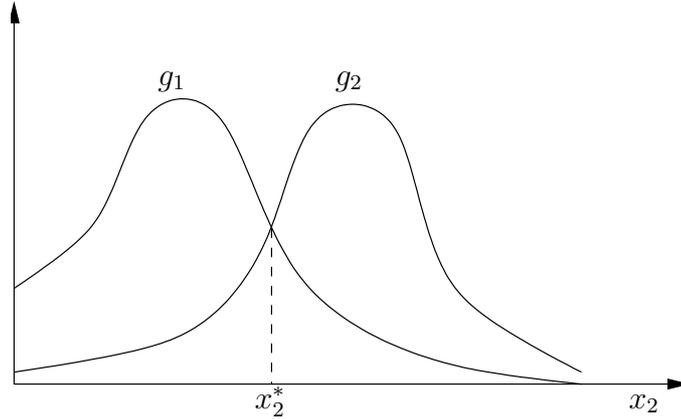


FIG. 1.6: Caractéristique de la boîte de vitesse

Le véhicule peut donc être modélisé par le système hybride suivant :

1. $\mathcal{Q} = \{1, 2\}$.
2. $\mathcal{E} = \{(1, 2), (2, 1)\}$.
3. $D_1 = D_2 = \mathbb{R}^2$.
4. $U_1 = U_2 = [0, u_{max}]$.
5. $f_q(x_1, x_2, u) = (x_2, ug_q(x_2) - kx_2)$, $q \in \mathcal{Q}$.
6. $G_{(1,2)} = G_{(2,1)} = \mathbb{R}^2$.
7. $R_{(1,2)}(x) = R_{(2,1)}(x) = \{x\}$.

1.2.4 Des systèmes à commutation aux systèmes hybrides

Un problème classique lié aux systèmes dynamiques à commutation consiste à déterminer une loi de rétroaction de la variable $x(t)$ sur le contrôle discret $q(t)$. Les changements de valeur de $q(t)$ ne sont plus décidés par l'extérieur mais déterminés par $x(t)$.

Dans le formalisme des systèmes hybrides, cela revient à spécifier les gardes du système qui détermineront les conditions de changement de la variable $q(t)$.

Exemple 1.2.4 (Boîte de vitesse automatique) Reprenons l'exemple 1.2.3, on veut équiper notre véhicule d'une boîte de vitesse automatique. Pour cela, on équipe le véhicule d'un ordinateur de bord. Celui-ci est informé en temps réel de la valeur de la vitesse du véhicule et détermine, en conséquence, le rapport le mieux adapté (voir figure 1.7).

La stratégie la plus simple et la plus facilement implémentable consiste à sélectionner le rapport le plus efficace; maximisant l'accélération du véhicule pour des apports de carburant égaux.

Si à un instant t ,

$$u(t)g_1(x_2(t)) - kx_2(t) \geq u(t)g_2(x_2(t)) - kx_2(t),$$

ou de manière équivalente,

$$g_1(x_2(t)) \geq g_2(x_2(t)),$$

alors le premier rapport est plus approprié. Donc, d'après la caractéristique de la boîte de vitesse (figure 1.6), si le véhicule à une allure inférieure à x_2^* , il vaut mieux choisir le premier rapport.

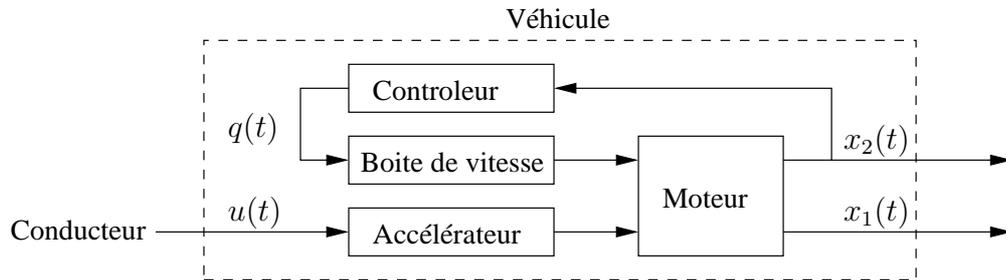


FIG. 1.7: Schéma du véhicule équipé d'une boîte de vitesse automatique

La figure 1.8 présente l'automate hybride modélisant le véhicule équipé d'un contrôleur implémentant cette stratégie.

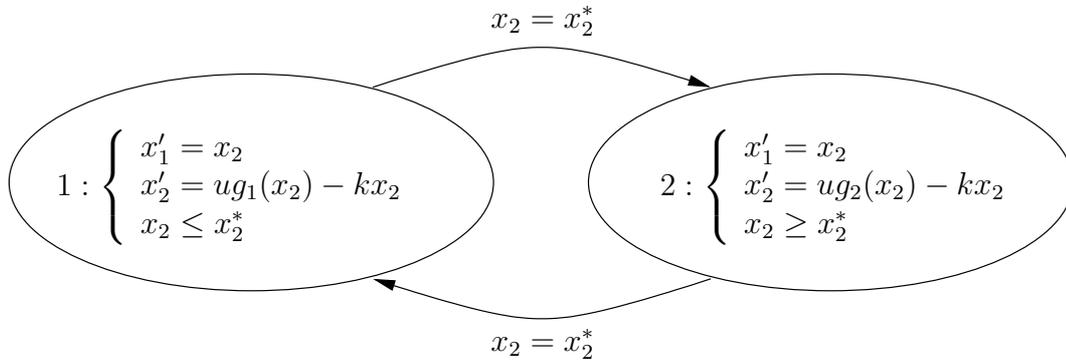


FIG. 1.8: Automate hybride de la boîte de vitesse automatique

Chapitre 2

Notion de Solution d'un Système Hybride

Dans le chapitre précédent, nous avons abordé, de manière intuitive, la notion de solution d'un système hybride. Dans cette partie, nous allons la formaliser. Les résultats exposés dans ce chapitre sont majoritairement inspirés par le travail de K.H. Johansson, J. Lygeros, S. Sastry et M. Egerstedt [64, 70].

2.1 Exécution d'un système hybride

2.1.1 Trajectoire temporisée

Nous avons vu que l'exécution d'un système hybride était caractérisée par l'évolution des variables $q(t)$ et $x(t)$. La variable discrète $q(t)$ est constante par morceaux et est donc entièrement donnée par les séquences $\{t_i\}_{i=0}^{i=N}$ des points de discontinuité et des valeurs successives de $q(t)$. La notion de trajectoire temporisée permet de décrire la séquence des instants de transition du système.

Définition 2.1.1 (Trajectoire temporisée) Une trajectoire temporisée $\tau = \{I_i\}_{i=0}^{i=N}$ est une séquence finie ou infinie d'intervalles de \mathbb{R} vérifiant

1. Pour $i < N$, $I_i = [t_i, t_{i+1}]$ avec $t_i \leq t_{i+1}$;
2. Si N est fini, $I_N = [t_N, t_{N+1}]$ avec $t_N \leq t_{N+1}$, ou $I_N = [t_N, t_{N+1}[$ avec $t_N < t_{N+1}$ (on peut avoir $t_{N+1} = +\infty$).

On notera, $t \in \tau$, si il existe $i \in \{0, \dots, N\}$, tel que $t \in I_i$.

On appelle longueur de la trajectoire temporisée la quantité $\sum_{i=0}^{i=N} (t_{i+1} - t_i)$. Une trajectoire temporisée de longueur finie (respectivement infinie) est dite limitée (respectivement illimitée).

Les trajectoires temporisées peuvent être séparées en quatre classes principales. En effet, soit τ une trajectoire temporisée, elle peut être :

- finie, limitée, $N < +\infty$, $t_{N+1} < +\infty$;
- finie, illimitée, $N < +\infty$, $t_{N+1} = +\infty$;
- infinie, limitée, $N = +\infty$, $\lim_{i \rightarrow +\infty} t_i < +\infty$;
- infinie, illimitée, $N = +\infty$, $\lim_{i \rightarrow +\infty} t_i = +\infty$.

Sur l'ensemble des trajectoires temporisées, on définit une relation d'ordre partiel.

Définition 2.1.2 (Préfixe) Soient $\tau = \{I_i\}_{i=0}^{i=N}$, $\tau' = \{J_i\}_{i=0}^{i=M}$ deux trajectoires temporisées. On dit que τ est un préfixe de τ' (noté $\tau \leq \tau'$) si elles sont identiques ou si τ est finie et

$$M \geq N, I_i = J_i, \text{ pour } i \in \{0, \dots, N-1\}, \text{ et } I_N \subseteq J_N.$$

Si de plus $\tau \neq \tau'$, on dit que τ est un préfixe strict de τ' (noté $\tau < \tau'$).

Il est clair que si τ est un préfixe strict d'une trajectoire temporisée, alors, nécessairement, τ est finie et limitée.

2.1.2 Trajectoire hybride

Ainsi, une trajectoire temporisée τ représente la séquence des intervalles sur lesquels les variables du système hybride $q(t)$ et $x(t)$ évoluent continûment.

Dans le premier chapitre, on a vu qu'une transition pouvait se produire à un instant t , lorsque $x(t)$ appartient à une garde. La transition se produit alors en temps nul; on affecte une nouvelle valeur à $q(t)$ et $x(t)$. A l'instant t où la transition s'effectue, les valeurs de $q(t)$ et $x(t)$ ne sont donc pas définies de manière unique. Pour lever l'ambiguïté, on définit la notion de trajectoire hybride (voir figure 2.1).

Définition 2.1.3 (Trajectoire hybride) Une trajectoire hybride dans un ensemble V est un doublet (τ, v) tel que :

1. $\tau = \{I_i\}_{i=0}^{i=N}$ est une trajectoire temporisée;
2. v est une séquence d'applications continues $\{v_i\}_{i=0}^{i=N}$, avec

$$\forall i \in \{0, \dots, N\}, v_i : I_i \rightarrow V .$$

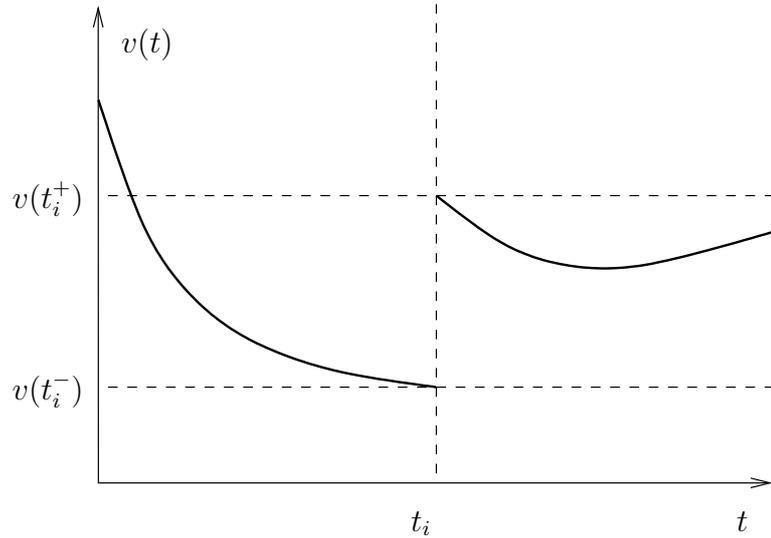


FIG. 2.1: Exemple d'une trajectoire hybride (τ, v)

Les bornes des intervalles de la trajectoire temporisée τ représentent les instants auxquels les transitions se produisent.

On note $v : \tau \rightarrow V$. Pour tout $t \in \tau$, tel que $t \notin \{t_1, \dots, t_N\}$, on définit

$$v(t) = v_i(t), \text{ si } t \in I_i .$$

Pour tout $t \in \{t_1, \dots, t_N\}$, $v(t)$ prend deux valeurs que l'on notera

$$v(t_i^-) = v_{i-1}(t_i), \quad v(t_i^+) = v_i(t_i) .$$

On définit sur l'ensemble des trajectoires hybrides une relation d'ordre partiel.

Définition 2.1.4 (Préfixe) Soient (τ, v) et (τ', v') deux trajectoires hybrides dans un ensemble V . On dit que (τ, v) est un préfixe de (τ', v') (noté $(\tau, v) \leq (\tau', v')$) si

$$\tau \leq \tau' \text{ et, pour tout } t \in \tau, v(t) = v'(t).$$

Si de plus, $(\tau, v) \neq (\tau', v')$, alors on dit que (τ, v) est un préfixe strict de (τ', v') (noté $(\tau, v) < (\tau', v')$).

Définition 2.1.5 (Plus grand commun préfixe) Soient (τ, v) et (τ', v') deux trajectoires hybrides, telles que $(t_0, v(t_0)) = (t'_0, v'(t_0))$. On appelle plus grand commun préfixe la trajectoire hybride $(\hat{\tau}, \hat{v})$ vérifiant :

1. $(\hat{\tau}, \hat{v}) \leq (\tau, v)$ et $(\hat{\tau}, \hat{v}) \leq (\tau', v')$;
2. $\forall (\bar{\tau}, \bar{v}), (\bar{\tau}, \bar{v}) \leq (\tau, v) \text{ et } (\bar{\tau}, \bar{v}) \leq (\tau', v') \implies (\bar{\tau}, \bar{v}) \leq (\hat{\tau}, \hat{v})$.

Pour deux trajectoires hybrides le plus grand commun préfixe représente donc leur plus grande partie commune avant que les deux trajectoires ne se séparent.

Proposition 2.1.1 Soient (τ, v) et (τ', v') deux trajectoires hybrides, soit $(\hat{\tau}, \hat{v})$ leur plus grand commun préfixe, alors :

- soit $\hat{\tau} = \tau$ ou $\hat{\tau} = \tau'$;
- soit $\hat{\tau} = \{\hat{I}_i\}_{i=0}^{\hat{N}}$, avec $\hat{N} < +\infty$ et $\hat{I}_{\hat{N}} = [\hat{t}_{\hat{N}}, \hat{t}_{\hat{N}+1}]$.

Preuve : Supposons $(\tau, v) \leq (\tau', v')$, alors $(\hat{\tau}, \hat{v}) = (\tau, v)$. De même, si $(\tau', v') \leq (\tau, v)$ alors $(\hat{\tau}, \hat{v}) = (\tau', v')$.

Supposons, maintenant que (τ, v) et (τ', v') ne sont pas comparables. Il est clair que $\hat{\tau} < \tau$ et $\hat{\tau} < \tau'$ et donc $\hat{\tau}$ est finie et limitée.

Donc $\hat{\tau} = \{\hat{I}_i\}_{i=0}^{\hat{N}}$, avec $\hat{N} < +\infty$. Si $\hat{I}_{\hat{N}} = [\hat{t}_{\hat{N}}, \hat{t}_{\hat{N}+1}[$, alors puisque $\hat{\tau}$ est un préfixe strict de τ et de τ' , on a $\hat{I}_{\hat{N}} \subset I_{\hat{N}}$ et $\hat{I}_{\hat{N}} \subset I'_{\hat{N}}$.

Par conséquent, $\hat{t}_{\hat{N}+1}$ appartient aux intervalles $I_{\hat{N}}$ et $I'_{\hat{N}}$.

De plus,

$$\forall t \in \hat{I}_{\hat{N}}, v_{\hat{N}}(t) = \hat{v}_{\hat{N}}(t) = v'_{\hat{N}}(t).$$

Par continuité de $v_{\hat{N}}$ sur $I_{\hat{N}}$ et de $v'_{\hat{N}}$ sur $I'_{\hat{N}}$.

$$v_{\hat{N}}(\hat{t}_{\hat{N}+1}) = \lim_{t \rightarrow \hat{t}_{\hat{N}+1}} v_{\hat{N}}(t) = \lim_{t \rightarrow \hat{t}_{\hat{N}+1}} v'_{\hat{N}}(t) = v'_{\hat{N}}(\hat{t}_{\hat{N}+1}).$$

On peut donc définir $(\bar{\tau}, \bar{v})$ le préfixe de (τ, v) et (τ', v') , tel que le dernier intervalle de $\bar{\tau}$ est $[\hat{t}_{\hat{N}}, \hat{t}_{\hat{N}+1}]$. On a de manière immédiate $(\hat{\tau}, \hat{v}) < (\bar{\tau}, \bar{v})$; $(\hat{\tau}, \hat{v})$ n'est donc pas le plus grand commun préfixe de (τ, v) et (τ', v') . ■

2.1.3 Définition d'une exécution d'un système hybride

Nous pouvons maintenant définir la notion de solution d'un système hybride (on parle d'exécution). L'ensemble des exécutions d'un système hybride est une partie de l'ensemble des trajectoires hybrides dans l'ensemble $\mathcal{Q} \times \mathbb{R}^n$.

Définition 2.1.6 (Exécution) Une exécution d'un système hybride \mathcal{H} est une trajectoire hybride (τ, q, x) , où $q : \tau \rightarrow \mathcal{Q}$ et $x : \tau \rightarrow \mathbb{R}^n$, vérifiant

1. **évolution continue** : pour tout $i \in \{0, \dots, N\}$, tel que $t_i < t_{i+1}$:
 - $q(t)$ est constante sur l'intervalle I_i ;
 - $x(t) \in D_{q(t)}$ sur l'intervalle I_i ;
 - il existe une application mesurable $u : [t_i, t_{i+1}] \rightarrow U_{q(t)}$, telle que

$$\forall t \in [t_i, t_{i+1}], x'(t) = f_{q(t)}(x(t), u(t)).$$

2. **évolution discrète** : pour tout $i \in \{1, \dots, N\}$:
 - $e = (q(t_i^-), q(t_i^+)) \in \mathcal{E}$;
 - $x(t_i^-) \in G_e$;
 - $x(t_i^+) \in R_e(x(t_i^-))$.

On dit qu'un système hybride accepte une exécution. Le triplet $(t_0, q(t_0), x(t_0))$ est appelé condition initiale de l'exécution.

Remarque 2.1.1 Pour toute condition initiale (t_0, q_0, x_0) , x_0 doit être un élément de D_{q_0} .

2.1.4 Classification des exécutions

Un système hybride peut accepter une grande variété d'exécutions. Une classification de ces exécutions se révèle utile.

Définition 2.1.7 (Exécution maximale) Une exécution (τ, q, x) est maximale si elle n'est le préfixe d'aucune autre exécution de \mathcal{H} .

Définition 2.1.8 (Exécution finie, infinie) Une exécution (τ, q, x) est dite :

- finie, si τ est finie et limitée;
- infinie, si τ est soit infinie, soit illimitée.

On a de manière assez immédiate le résultat suivant.

Proposition 2.1.2 Une exécution infinie est maximale.

Preuve : Soit (τ, q, x) une exécution infinie. τ est soit infinie, soit illimitée et ne peut donc pas être le préfixe d'une autre trajectoire temporisée. Par conséquent, l'exécution (τ, q, x) n'est le préfixe d'aucune autre trajectoire hybride et est donc maximale. ■

Exemple 2.1.1 On considère le système hybride \mathcal{H} défini par :

1. $\mathcal{Q} = \{1, 2\}$
2. $\mathcal{E} = \{(1, 2), (2, 1)\}$
3. $D_1 = D_2 = [-3, 3]$
4. $f_1(x) = 1, f_2(x) = -1$
5. $G_{(1,2)} = [1, 2], G_{(2,1)} = [-2, -1]$
6. $R_{(1,2)} = R_{(2,1)} = \{x\}$

\mathcal{H} accepte les trois exécutions suivantes (voir figure 2.2).

Première exécution : (τ_1, q_1, x_1) avec

- $\tau_1 = \{[2i, 2i + 2]\}_{i=0}^{i=+\infty}$;
- Pour tout $t \in [2i, 2i + 2]$, $q_1(t) = \begin{cases} 1, & \text{si } i \text{ est pair} \\ 2, & \text{si } i \text{ est impair;} \end{cases}$
- Pour tout $t \in [2i, 2i + 2]$, $x_1(t) = \begin{cases} -1 + (t - 2i), & \text{si } i \text{ est pair} \\ 1 - (t - 2i), & \text{si } i \text{ est impair.} \end{cases}$

τ_1 est infinie et illimitée, par conséquent, l'exécution (τ_1, q_1, x_1) est infinie et donc maximale.

Deuxième exécution : (τ_2, q_2, x_2) avec

- $\tau_2 = \{[0, 2], [2, 4]\}$;
- Pour tout $t \in \tau_2$, $q_2(t) = \begin{cases} 1, & \text{si } t \in [0, 2] \\ 2, & \text{si } t \in [2, 4]; \end{cases}$
- Pour tout $t \in \tau_2$, $x_2(t) = \begin{cases} -1 + t, & \text{si } t \in [0, 2] \\ 3 - t, & \text{si } t \in [2, 4]. \end{cases}$

τ_2 est finie et limitée, par conséquent l'exécution (τ_2, q_2, x_2) est finie. De plus, il est clair que $(\tau_2, q_2, x_2) < (\tau_1, q_1, x_1)$ et donc (τ_2, q_2, x_2) n'est pas maximale.

Troisième exécution : (τ_3, q_3, x_3) avec

- $\tau_3 = \{[0, 2], [2, 6]\}$;
- Pour tout $t \in \tau_3$, $q_3(t) = \begin{cases} 1, & \text{si } t \in [0, 2] \\ 2, & \text{si } t \in [2, 6]; \end{cases}$
- Pour tout $t \in \tau_3$, $x_3(t) = \begin{cases} -1 + t, & \text{si } t \in [0, 2] \\ 3 - t, & \text{si } t \in [2, 6]. \end{cases}$

τ_3 est finie et limitée, par conséquent l'exécution (τ_3, q_3, x_3) est finie. De plus, (τ_3, q_3, x_3) est maximale.

En effet, supposons qu'il existe une exécution (τ, q, x) , telle que $(\tau_3, q_3, x_3) < (\tau, q, x)$. Soit le deuxième intervalle de la trajectoire temporisée τ est de la forme $[2, T]$ (ou $[2, T[$), avec $T > 6$. Soit la trajectoire temporisée τ possède un troisième intervalle.

Dans le premier cas, pour tout $t \in [2, T[$, $q(t) = 2$, et $x(t) = 3 - t$. Par conséquent, pour $t \in]6, T[$, $x(t) < -3$, et donc $x(t) \notin D_2$.

Dans le deuxième cas, cela signifie qu'il y a un changement de l'état discret à l'instant $t = 6$, On a donc $q(6^-) = 2$ et $q(6^+) = 1$. Or $x(6^-) = -3$, et donc $x(6^-) \notin G_{(2,1)}$.

On voit donc que la réciproque de la proposition 2.1.2 est fautive : une exécution maximale peut ne pas être infinie.

Notons que (τ_2, q_2, x_2) est aussi un préfixe strict de (τ_3, q_3, x_3) ; c'est d'ailleurs le plus grand commun préfixe de (τ_1, q_1, x_1) et (τ_3, q_3, x_3) . Une exécution peut donc être le préfixe de plusieurs exécutions maximales distinctes.

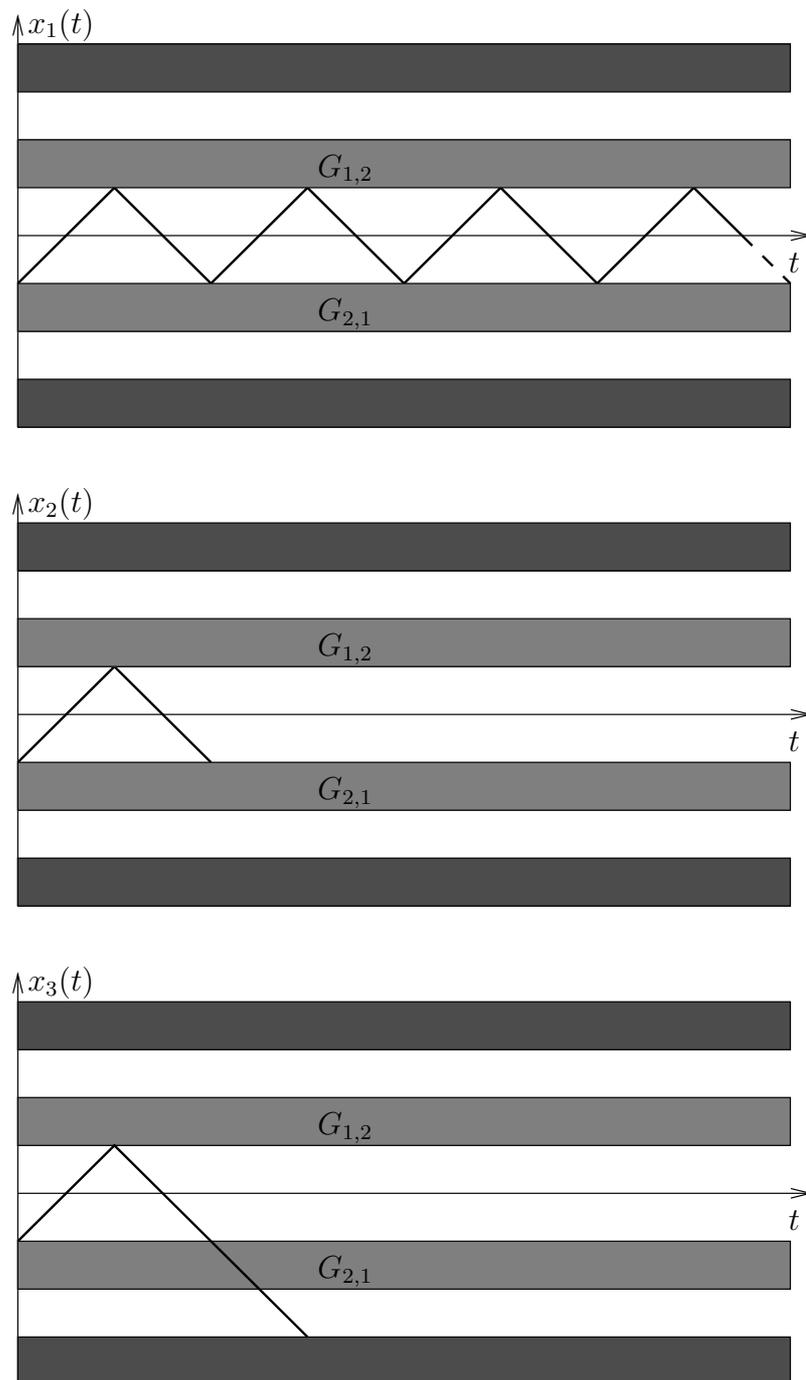


FIG. 2.2: Exemple d'exécutions

Cet exemple suscite deux questions :

1. Comment garantir, que toute exécution est le préfixe d'une seule exécution maximale?
2. Sous quelles conditions la réciproque de la proposition 2.1.2 est elle vraie? Autrement dit, quelles conditions doivent être vérifiées pour garantir que toute exécution finie

est le préfixe d'une exécution infinie?

Nous examinons ces deux questions dans la partie suivante.

2.2 Systèmes hybrides déterministes et non-bloquants

Les deux questions précédentes nous amènent à définir deux classes de systèmes hybrides, les systèmes hybrides déterministes et les systèmes hybrides non-bloquants.

Définition 2.2.1 (Système hybride déterministe) *Un système hybride \mathcal{H} est déterministe, si, pour toute condition initiale (t_0, q_0, x_0) , \mathcal{H} accepte une unique exécution maximale.*

Définition 2.2.2 (Système hybride non-bloquant) *Un système hybride \mathcal{H} est non-bloquant, si, pour toute condition initiale (t_0, q_0, x_0) , \mathcal{H} accepte une exécution infinie.*

Dans cette partie, on suppose que les équations différentielles définissant la dynamique continue du système sont autonomes. On considère donc un système hybride $\mathcal{H} = \{\mathcal{Q}, \mathcal{E}, \mathcal{D}, \mathcal{F}, \mathcal{G}, \mathcal{R}\}$.

On suppose de plus, que pour tout $q \in \mathcal{Q}$, le champ de vecteur f_q est Lipschitz sur D_q . Ainsi [30], pour tout $x_0 \in D_q$, le problème de Cauchy

$$x'(t) = f_q(x(t)), x(t_0) = x_0, x(t) \in D_q \quad (2.2.1)$$

admet une unique solution maximale. Cette solution est définie sur un intervalle I qui peut être de la forme

- $I = [t_0, +\infty[$;
- $I = [t_0, T]$, avec $T < +\infty$; dans ce cas, $x(T) \in \partial D_q$;
- $I = [t_0, T[$, avec $T < +\infty$; dans ce cas, la limite de $x(t)$ en T existe et

$$\lim_{t \rightarrow T} x(t) \in \partial D_q.$$

L'ensemble de ces points de ∂D_q jouent un rôle essentiel pour les propriétés de déterminisme ou de non-blocage du système hybride. D'une part, si une transition du système hybride est possible alors que $x(t)$ n'est pas un point de cet ensemble, le système hybride peut soit continuer à évoluer continûment soit effectuer une transition. D'autre part, si lorsque $x(t)$ atteint un de ces points, toute transition est impossible, alors l'exécution du système hybride ne peut se poursuivre.

Définition 2.2.3 (Point de sortie) *Soit x_0 un point de D_q , tel que $x(t)$, la solution maximale du problème de Cauchy (2.2.1) est définie sur un intervalle I de longueur finie.*

- Si $I = [t_0, T]$, alors $x(T)$ est un point de sortie de D_q .
- Si $I = [t_0, T[$, alors $\lim_{t \rightarrow T} x(t)$ est un point de sortie de D_q .

L'ensemble des points de sortie est une partie de ∂D_q notée S_q .

A l'aide des ensembles des points de sortie, il est possible de donner une caractérisation d'un système hybride déterministe ou non-bloquant [70].

2.2.1 Caractérisation d'un système hybride déterministe

Lemme 2.2.1 (Condition nécessaire et suffisante [70]) *Un système hybride est déterministe si et seulement si il vérifie les trois conditions suivantes :*

- pour tout $(q, q') \in \mathcal{E}$, $G_{(q, q')} \subseteq S_q$;
- pour tout $(q, q') \in \mathcal{E}$ et $(q, q'') \in \mathcal{E}$, $q' \neq q''$, $G_{(q, q')} \cap G_{(q, q'')} = \emptyset$;
- pour tout $e \in \mathcal{E}$ et $x \in G_e$, $R_e(x)$ contient un unique élément.

Pratiquement, pour qu'un système hybride soit déterministe il faut et il suffit qu'une transition ne soit possible que lorsque l'évolution continue est bloquée. De plus, lorsqu'une transition est autorisée, toutes les autres sont bloquées et la variable continue ne peut être réinitialisée qu'à une seule valeur.

2.2.2 Caractérisation d'un système hybride non-bloquant

Lemme 2.2.2 (Condition suffisante [70]) *Si*

$$\forall q \in \mathcal{Q}, \forall x \in S_q, \exists (q, q') \in \mathcal{E}, x \in G_{(q, q')},$$

alors le système hybride est non-bloquant.

De plus, toute exécution maximale acceptée par \mathcal{H} est infinie.

De manière intuitive, la condition donnée dans le lemme 2.2.2 signifie qu'à chaque fois que l'évolution continue du système est bloquée (quand la variable continue atteint la frontière du domaine et est obligée de sortir de celui-ci) alors une transition est possible.

Cette condition n'est pas nécessaire. En effet, considérons le système hybride \mathcal{H} donné par

1. $\mathcal{Q} = \{1, 2\}$
2. $\mathcal{E} = \{(1, 2)\}$
3. $D_1 = [0, 1[, D_2 = \mathbb{R}$
4. $f_1(x) = 1, f_2(x) = 0$
5. $G_{(1, 2)} = [0, 1[$
6. $R_{(1, 2)} = \{x\}$

L'ensemble des points de sorties du domaine D_1 est $S_1 = \{1\}$, donc les hypothèses du lemme 2.2.2 ne sont pas vérifiées. Soit (t_0, σ_0, y_0) une condition initiale.

Si $\sigma_0 = 2$, alors l'exécution

$$(\{[t_0, +\infty[, \{t \mapsto 2\}, \{t \mapsto y_0\})$$

est infinie et acceptée par \mathcal{H} .

Si $\sigma_0 = 1$ alors l'exécution

$$(\{[t_0, t_0], [t_0, +\infty[, \{t \mapsto 1, t \mapsto 2\}, \{t \mapsto y_0, t \mapsto y_0\})$$

est infinie et acceptée par \mathcal{H} .

Donc, pour toute condition initiale, le système hybride accepte une exécution infinie; \mathcal{H} est non-bloquant.

En revanche, pour un système hybride déterministe, la condition du lemme 2.2.2 devient nécessaire.

Lemme 2.2.3 (Condition nécessaire et suffisante [70]) *Un système hybride déterministe est non-bloquant si et seulement si*

$$\forall q \in \mathcal{Q}, \forall x \in S_q, \exists (q, q') \in \mathcal{E}, x \in G_{(q, q')}.$$

2.2.3 Existence et unicité des exécutions

Grâce aux lemmes 2.2.1 et 2.2.3, on peut donner des conditions nécessaires et suffisantes pour l'existence et l'unicité d'une exécution infinie (et donc maximale) acceptée par le système hybride.

Théorème 2.2.1 (Existence et unicité des exécutions [70]) *Soit un système hybride \mathcal{H} vérifiant les trois conditions suivantes*

- pour tout $q \in \mathcal{Q}$ $\bigcup_{(q, q') \in \mathcal{E}} G_{(q, q')} = S_q$;
- pour tout $(q, q') \in \mathcal{E}$ et $(q, q'') \in \mathcal{E}$, $G_{(q, q')} \cap G_{(q, q'')} = \emptyset$;
- pour tout $e \in \mathcal{E}$ et $x \in G_e$, $R_e(x)$ contient un unique élément.

Alors, pour toute condition initiale (t_0, σ_0, y_0) , \mathcal{H} accepte une unique exécution infinie (τ, q, x) . Toute exécution de \mathcal{H} ayant (t_0, σ_0, y_0) pour condition initiale est un préfixe de (τ, q, x) .

Preuve : D'après le lemme 2.2.3, \mathcal{H} est non-bloquant. Par conséquent, pour toute condition initiale (t_0, σ_0, y_0) il existe une exécution infinie (τ, q, x) .

D'après le lemme 2.2.1, \mathcal{H} est déterministe. Donc, pour toute condition initiale, il existe une unique exécution maximale.

D'après la proposition 2.1.2, toute exécution infinie est maximale; donc, pour toute condition initiale, il existe une unique exécution infinie.

De plus, comme \mathcal{H} est déterministe, toute exécution ayant (t_0, σ_0, y_0) pour condition initiale est un préfixe de (τ, q, x) . ■

Ainsi, l'existence et l'unicité d'une exécution infinie pour toute condition initiale est garantie par des critères simples.

2.3 Exécution Zénon

On suppose que les hypothèses du théorème 2.2.1 sont vérifiées. Ainsi, pour toute condition initiale, le système hybride \mathcal{H} accepte une unique exécution infinie (τ, q, x) . La trajectoire temporisée τ peut donc être :

- finie illimitée, le système hybride effectue un nombre fini de transitions en un temps infini;
- infinie illimitée, le système hybride effectue une infinité de transitions en un temps infini;
- infinie limitée, le système hybride effectue une infinité de transitions en un temps fini.

Dans le dernier cas, on dit que l'exécution (τ, q, x) est Zénon.

Définition 2.3.1 (Exécution Zénon) *Une exécution (τ, q, x) est Zénon si τ est infinie et limitée.*

Exemple 2.3.1 (Balle bondissante) Reprenons l'exemple 1.2.1 du premier chapitre. A l'instant t_0 , on laisse tomber la balle d'une altitude $z_0 > 0$. La variable continue du système est $x = (x_1, x_2)$, où x_1 est l'altitude de la balle et x_2 sa vitesse. La condition initiale du système hybride est donc $(t_0, q, (z_0, 0))$. Tant que l'altitude de la balle est positive, on a :

$$x_1'(t) = x_2(t) \text{ et } x_2'(t) = -g \implies x_2(t) = -g(t - t_0) \text{ et } x_1(t) = -\frac{g}{2}(t - t_0)^2 + z_0.$$

L'impact de la balle sur le sol (et donc la transition du système hybride) se produit à l'instant t_1 tel que $x_1(t_1) = 0$, donc

$$t_1 = t_0 + \sqrt{\frac{2z_0}{g}}, \quad x_2(t_1^-) = -\sqrt{2gz_0}.$$

A cet instant, la balle rebondit :

$$x_2(t_1^+) = -cx_2(t_1^-) = c\sqrt{2gz_0}.$$

Soit t_i l'instant auquel se produit le i -ème rebond. Jusqu'au prochain impact de la balle sur le sol, on a

$$x_2(t) = -g(t - t_i) + x_2(t_i^+) \text{ et } x_1(t) = -\frac{g}{2}(t - t_i)^2 + x_2(t_i^+)(t - t_i).$$

A l'instant t_{i+1} , la balle touche le sol

$$x_1(t_{i+1}) = 0 \implies t_{i+1} = t_i + \frac{2}{g}x_2(t_i^+)$$

et rebondit

$$x_2(t_{i+1}^+) = -cx_2(t_{i+1}^-) = cx_2(t_i^+) = \dots = c^i x_2(t_1^+) = c^{i+1} \sqrt{2gz_0}.$$

Ainsi, le système hybride admet une exécution infinie et la longueur de la trajectoire temporisée associée est

$$\sum_{i=0}^{i=\infty} (t_{i+1} - t_i) = \sqrt{\frac{2z_0}{g}} + \sum_{i=1}^{i=\infty} \frac{2}{g} x_2(t_i^+) = \sqrt{\frac{2z_0}{g}} + \sum_{i=1}^{i=\infty} c^i 2 \sqrt{\frac{2z_0}{g}}.$$

Si $c < 1$, alors

$$\sum_{i=0}^{i=\infty} (t_{i+1} - t_i) = \sqrt{\frac{2z_0}{g}} \left(1 + \frac{2c}{1-c} \right) < +\infty.$$

La trajectoire temporisée est donc limitée et l'exécution est Zénon.

L'existence d'exécutions Zénon pose parfois des problèmes pour la simulation des systèmes hybrides. Les instants de transition du système hybride se rapprochent de plus en plus. Si l'on procède à une simulation numérique, on ne peut donc plus, à partir d'un certain rang, différencier les instants successifs t_i .

Hormis le problème de simulation, l'existence d'exécutions Zénon pose un problème plus conceptuel. En effet, si l'on reprend l'exemple précédent, on constate que l'on ne peut prévoir l'évolution de la balle que sur un intervalle de temps limité. Or, passé ce délai, la balle continue à exister. Y-a-t-il un moyen de savoir ce qui se passe après?

Remarquons d'abord que la possibilité d'avoir une exécution Zénon est liée à un des points essentiels de la définition des systèmes hybrides : la transition en temps nul. En effet, il est clair que ce fait relève de l'idéalisation du monde réel. Pour une balle réelle, une durée non-nulle sépare l'instant où la balle rencontre le sol et l'instant où elle le quitte.

Une manière de *régulariser* une exécution Zénon est donc de supposer que les transitions du système ne se font plus en temps nul mais se déroulent en une durée ε constante. Aussi petit le délai ε soit-il, les variables du système hybride sont alors définies sur $[t_0, +\infty[$. On peut alors prévoir l'évolution de la variable continue après le temps Zénon (i.e. $\lim_{i \rightarrow +\infty} t_i$) en prenant la limite de la variable $x(t)$ lorsque le paramètre ε tend vers 0.

L'exécution ainsi régularisée du système hybride modélisant la balle nous informe que celle-ci atteint le point $(0, 0)$ au temps Zénon et reste immobile par la suite.

Cette méthode de régularisation et d'autres, telle que la généralisation du concept de solution Filippov (voir [41]) pour une équation différentielle discontinue au concept d'exécution Filippov d'un système hybride, ont été abordées dans [64, 65, 74]. Des conditions nécessaires ou suffisantes à l'existence d'exécutions Zénon peuvent être trouvées dans [64, 100].

Chapitre 3

Systemes Hybrides Linéaires par Morceaux

La classe des systèmes hybrides linéaires par morceaux est sans aucun doute l'une des plus utilisées. En effet, elle présente de nombreux avantages et notamment :

- elle permet de modéliser de manière assez réaliste de nombreux problèmes (en biologie [12, 33, 44], en électronique [57, 83]...);
- elle est suffisamment simple pour pouvoir concevoir des outils algorithmiques efficaces pour l'analyse [40, 52, 63, 77] ou le contrôle [3, 14, 68].

3.1 Définition du modèle hybride linéaire par morceaux

Dans cette partie, nous présentons les restrictions du modèle linéaire par morceaux par rapport au modèle général de la définition 1.1.1.

La variable discrète du système prend ses valeurs dans un ensemble dénombrable \mathcal{Q} . Sa dynamique, comme dans le modèle général, est décrite par l'automate $(\mathcal{Q}, \mathcal{E})$ où \mathcal{E} est l'ensemble des transitions discrètes du système. Pour tout $q \in \mathcal{Q}$, on définit le sous-ensemble E_q de \mathcal{E} des transitions de q vers un autre état :

$$E_q = \{e \in \mathcal{E}, e = (q, q')\}.$$

A chaque état $q \in \mathcal{Q}$, on associe une équation différentielle linéaire qui peut être :

- autonome, dans ce cas, la dynamique de la variable continue est donnée par

$$x'(t) = A_q x(t) + b_q$$

où A_q est une matrice $n \times n$ à coefficients constants et b_q est un vecteur constant de \mathbb{R}^n ;

- avec entrée, dans ce cas,

$$x'(t) = A_q x(t) + B_q u(t), u(t) \in U_q$$

où A_q et B_q sont des matrices à coefficients constants de tailles respectives $n \times n$ et $n \times p$ et U_q est un polyèdre de \mathbb{R}^p .

Pour tout $q \in \mathcal{Q}$, on associe également un ensemble de contraintes. Chaque contrainte correspond à une transition de l'ensemble E_q . La contrainte C_e , associée à la transition e , peut s'écrire sous la forme d'une inégalité linéaire, $C_e = \{x \in \mathbb{R}^n, c_e^T x \geq d_e\}$, où c_e est un vecteur unitaire de \mathbb{R}^n et d_e un scalaire. Lorsque la variable discrète vaut q , la variable continue $x(t)$ doit satisfaire l'ensemble des contraintes :

$$\forall e \in E_q, c_e^T x(t) \geq d_e.$$

Le domaine D_q est donc défini par

$$D_q = \bigcap_{e \in E_q} C_e = \{x \in \mathbb{R}^n, \forall e \in E_q, c_e^T x \geq d_e\}.$$

D_q est donc un polyèdre (voir figure 3.1).

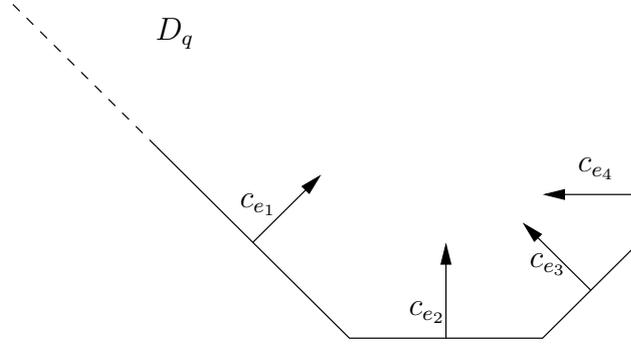


FIG. 3.1: Exemple d'un polyèdre D_q

Quand $x(t)$ est sur le point de franchir l'hyperplan d'équation $c_e^T x = d_e$, alors la transition e s'effectue. Les gardes du système sont donc définies par la relation

$$\forall e \in E_q, G_e = S_q \cap \{x \in \mathbb{R}^n, c_e^T x = d_e\}$$

où S_q est l'ensemble des points de sortie de D_q . Pour les systèmes autonomes, S_q est donné par la définition 2.2.3. On peut étendre la définition de S_q aux systèmes avec entrée de la manière suivante : $y \in \partial D_q$ est un élément de S_q , si il existe une solution $x(t)$ de l'équation différentielle associée au champ de vecteurs f_q telle que $x(0) = y$ et

$$\exists \varepsilon > 0, \forall t \in]0, \varepsilon[, x(t) \notin D_q.$$

Enfin, on suppose qu'il n'y a pas de réinitialisation de la variable continue :

$$\forall e \in \mathcal{E}, R_e(x) = \{x\}.$$

Remarque 3.1.1 La dernière hypothèse implique que

$$\forall e = (q, q') \in \mathcal{E}, G_e \subseteq D_{q'}.$$

En effet, R_e est une application de G_e dans l'ensemble des parties de $D_{q'}$. Donc, pour tout $x \in G_e$, on doit avoir $\{x\} \subseteq D_{q'}$.

Un système hybride linéaire par morceaux peut donc se définir de la manière suivante.

Définition 3.1.1 (Système hybride linéaire par morceaux) *Un système hybride linéaire par morceaux est un quintuplet*

$$\mathcal{H} = (\mathcal{Q}, \mathcal{E}, \mathcal{U}, \mathcal{F}, \mathcal{C})$$

où :

1. \mathcal{Q} est l'ensemble dénombrable des états discrets (ou modes).
2. $\mathcal{E} \subseteq \mathcal{Q} \times \mathcal{Q}$ est l'ensemble des arêtes (ou transitions).
3. $\mathcal{U} = \{U_q, q \in \mathcal{Q}\}$ est la collection des domaines de contrôle.
 $\forall q \in \mathcal{Q}, U_q$ est un polyèdre de \mathbb{R}^p .
4. $\mathcal{F} = \{f_q, q \in \mathcal{Q}\}$ est la collection des champs de vecteurs linéaires.
 $\forall q \in \mathcal{Q}, f_q(x, u) = A_q x + B_q u$ ou $f_q(x) = A_q x + b_q$.
5. $\mathcal{C} = \{C_e, e \in \mathcal{E}\}$ est la collection des contraintes.
 $\forall e \in \mathcal{E}, C_e = \{x \in \mathbb{R}^n, c_e^T x \geq d_e\}$.

3.2 Discussion sur l'existence et l'unicité des exécutions

3.2.1 Systèmes autonomes

On suppose, dans ce paragraphe, que les champs de vecteurs du système hybride \mathcal{H} définissent des équations différentielles autonomes. Dans ce cas, $\mathcal{H} = (\mathcal{Q}, \mathcal{E}, \mathcal{F}, \mathcal{C})$.

Tout d'abord, l'application directe du lemme 2.2.2 nous donne le résultat suivant.

Proposition 3.2.1 *Un système hybride linéaire par morceaux \mathcal{H} avec des dynamiques continues autonomes est non-bloquant. Toute exécution maximale acceptée par \mathcal{H} est infinie.*

Preuve : Soient $q \in \mathcal{Q}$ et $x \in S_q$. Puisque $S_q \subseteq \partial D_q$, il existe une transition $e \in E_q$ telle que $c_e^T x = d_e$ et donc $x \in G_e$. ■

Généralement, un système hybride linéaire par morceaux n'est pas déterministe. Supposons qu'il existe un état $q \in \mathcal{Q}$ et deux transitions partant de q , $e_1 = (q, q_1)$ et $e_2 = (q, q_2)$, tels que

$$S_q \cap \{x \in \mathbb{R}^n, c_{e_1}^T x - d_{e_1}\} \cap \{x \in \mathbb{R}^n, c_{e_2}^T x - d_{e_2}\} \neq \emptyset.$$

Soit y un point de cet ensemble, pour la condition initiale $(0, q, y)$, le système hybride accepte les deux exécutions :

$$\begin{aligned} & (\{[0, 0], [0, 0]\}, \{t \mapsto q, t \mapsto q_1\}, \{t \mapsto y, t \mapsto y\}) \\ \text{et} & (\{[0, 0], [0, 0]\}, \{t \mapsto q, t \mapsto q_2\}, \{t \mapsto y, t \mapsto y\}). \end{aligned}$$

Cependant, cette configuration est la seule qui puisse introduire de l'indéterminisme.

Proposition 3.2.2 *Supposons que \mathcal{H} accepte deux exécutions maximales distinctes (τ, q, x) et (τ', q', x') pour la même condition initiale. Soit $(\hat{\tau}, \hat{q}, \hat{x})$ leur plus grand commun préfixe, on note $(\sigma, y) = (\hat{q}(\hat{t}_{\hat{N}+1}), \hat{x}(\hat{t}_{\hat{N}+1}))$. Il existe deux transitions e_1, e_2 dans E_σ , $e_1 \neq e_2$, telles que*

$$\begin{cases} y \in S_\sigma \\ c_{e_1}^T y - d_{e_1} = 0 \\ c_{e_2}^T y - d_{e_2} = 0. \end{cases}$$

Preuve : Comme (τ, q, x) et (τ', q', x') sont distinctes et maximales, alors leur plus grand commun préfixe $(\hat{\tau}, \hat{q}, \hat{x})$ vérifie :

$$(\hat{\tau}, \hat{q}, \hat{x}) < (\tau, q, x) \text{ et } (\hat{\tau}, \hat{q}, \hat{x}) < (\tau', q', x').$$

On a donc $\hat{\tau} \neq \tau$ et $\hat{\tau} \neq \tau'$. D'après la proposition 2.1.1, $\hat{\tau} = \{\hat{I}_i\}_{i=0}^{\hat{N}}$, avec $\hat{N} < +\infty$ et $\hat{I}_{\hat{N}} = [\hat{t}_{\hat{N}}, \hat{t}_{\hat{N}+1}]$. On note $(\sigma, y) = (\hat{q}(\hat{t}_{\hat{N}+1}), \hat{x}(\hat{t}_{\hat{N}+1}))$.

–**Premier cas :** $\hat{t}_{\hat{N}+1} \neq t_{\hat{N}+1}$ ou $\hat{t}_{\hat{N}+1} \neq t'_{\hat{N}+1}$

A l'instant $\hat{t}_{\hat{N}+1}$, il n'y a pas de transitions dans aucune des deux exécutions maximales. Cependant, $x(\hat{t}_{\hat{N}+1}) = x'(\hat{t}_{\hat{N}+1}) = y$, donc par unicité de la solution du problème de Cauchy, $x(t)$ et $x'(t)$ coïncident sur un intervalle plus grand que $\hat{I}_{\hat{N}}$. $(\hat{\tau}, \hat{q}, \hat{x})$ n'est donc pas le plus grand commun préfixe de (τ, q, x) et (τ', q', x') .

–**Deuxième cas :** $\hat{t}_{\hat{N}+1} = t_{\hat{N}+1}$ et $\hat{t}_{\hat{N}+1} \neq t'_{\hat{N}+1}$

A l'instant $\hat{t}_{\hat{N}+1}$, il y a une transition dans l'exécution (τ, q, x) donc y appartient à une garde du domaine D_σ et donc à S_σ .

Par contre, il n'y a pas de transition dans l'exécution (τ', q', x') , donc y n'est pas un élément de S_σ . Ce cas est impossible.

–**Troisième cas :** $\hat{t}_{\hat{N}+1} \neq t_{\hat{N}+1}$ et $\hat{t}_{\hat{N}+1} = t'_{\hat{N}+1}$, symétrique au deuxième cas.

–**Quatrième cas :** $\hat{t}_{\hat{N}+1} = t_{\hat{N}+1}$ et $\hat{t}_{\hat{N}+1} = t'_{\hat{N}+1}$

A l'instant $\hat{t}_{\hat{N}+1}$, il y a une transition dans les deux exécutions. Cependant, comme $(\hat{\tau}, \hat{q}, \hat{x})$ est le plus grand commun préfixe de (τ, q, x) et (τ', q', x') ,

$$(q(t_{\hat{N}+1}^+), x(t_{\hat{N}+1}^+)) \neq (q'(t_{\hat{N}+1}^+), x'(t_{\hat{N}+1}^+)).$$

Par hypothèse, il n'y a pas de réinitialisation de la variable continue, donc

$$x(t_{\hat{N}+1}^+) = x(t_{\hat{N}+1}^-) = y = x'(t_{\hat{N}+1}^-) = x'(t_{\hat{N}+1}^+).$$

Nécessairement $q(t_{\hat{N}+1}^+) \neq q'(t_{\hat{N}+1}^+)$. Donc $y \in G_{(\sigma, q(t_{\hat{N}+1}^+))} \cap G_{(\sigma, q'(t_{\hat{N}+1}^+))}$.

■

3.2.2 Systèmes avec entrée

On suppose maintenant que les champs de vecteurs du système hybride \mathcal{H} définissent des équations différentielles avec entrée.

Proposition 3.2.3 *Un système hybride linéaire par morceaux est non-bloquant. Toute exécution maximale acceptée par \mathcal{H} est infinie.*

Preuve : Soit

$$\mathcal{H} = (\mathcal{Q}, \mathcal{E}, \mathcal{U}, \mathcal{F}, \mathcal{C})$$

un système hybride linéaire par morceaux. Pour tout $q \in \mathcal{Q}$, prenons un vecteur u_q du polyèdre U_q . Considérons le système hybride linéaire par morceaux avec des dynamiques continues autonomes

$$\mathcal{H}' = (\mathcal{Q}, \mathcal{E}, \mathcal{F}', \mathcal{C}), \text{ avec } \forall q \in \mathcal{Q}, f_q(x) = A_q x + B_q u_q .$$

Toute exécution acceptée par \mathcal{H}' l'est également par \mathcal{H} . D'après le lemme 3.2.1, toute exécution maximale de \mathcal{H}' est infinie.

Supposons qu'il existe (τ, q, x) une exécution maximale et finie de \mathcal{H} . On a $\tau = \{I_i\}_{i=0}^{i=N}$ avec $N < +\infty$. Il est facile de montrer que $I_N = [t_N, t_{N+1}]$ (preuve similaire à celle du lemme 2.1.1) .

Or \mathcal{H}' accepte une exécution infinie pour la condition initiale $(t_{N+1}, q(t_{N+1}), x(t_{N+1}))$. Cette exécution est également une exécution de \mathcal{H} et prolonge (τ, q, x) qui ne peut donc pas être maximale. ■

Pour une condition initiale donnée, une équation différentielle linéaire avec entrée admet plusieurs solutions. Par conséquent, il est clair que les systèmes avec entrée ne sont, en général, pas déterministes.

Par la suite, nous ne considérerons que des systèmes hybrides linéaires par morceaux.

Deuxième partie

Calcul des Exécutions d'un
Système Hybride

Chapitre 4

Simulation des Systèmes Hybrides

Soient $\mathcal{H} = (\mathcal{Q}, \mathcal{E}, \mathcal{F}, \mathcal{C})$ un système hybride linéaire par morceaux autonome et (t_0, σ_0, y_0) une condition initiale de \mathcal{H} . La simulation de \mathcal{H} consiste à calculer une exécution (τ, q, x) acceptée par \mathcal{H} et ayant (t_0, σ_0, y_0) pour condition initiale. Dans ce chapitre, nous présentons un algorithme de simulation de \mathcal{H} .

Contrairement à l'approche classique de la simulation des systèmes continus (voir par exemple [29, 32]) qui consiste, le plus souvent, à calculer la valeur des variables du système à certains instants de discrétisation, notre approche cherche à fournir une représentation formelle de l'exécution.

Ainsi, l'exécution de \mathcal{H} sera représentée, du point de vue informatique, par des séquences d'intervalles et d'applications, ce qui permet de rester au plus près de la définition 2.1.6.

Cette représentation est possible, car les champs de vecteurs que nous considérons sont linéaires et donc une résolution formelle des équations différentielles associées est possible.

Cette approche possède deux avantages :

1. Elle permet de représenter n'importe quelle exécution (τ, q, x) où τ est une trajectoire temporisée finie. On peut donc avoir des exécutions définies sur $[t_0, +\infty[$.
2. Elle permet d'évaluer les variables q et x de \mathcal{H} à tout instant de l'intervalle couvert par la trajectoire temporisée τ .

Cependant, nous sommes obligés de ne considérer que des trajectoires temporisées finies. En effet, une représentation de séquences infinies n'est évidemment pas envisageable. Nous devons donc fixer un nombre maximal de transitions autorisées que l'on notera $N < +\infty$.

Nous pouvons alors proposer l'algorithme de simulation de \mathcal{H} . Il consiste en une boucle principale; à chaque itération de la boucle, les séquences τ , q et x sont augmentées d'un élément. Ainsi, chaque itération de la boucle consiste à simuler ce qui se passe entre deux transitions discrètes. Chaque itération se décompose en trois phases (voir également [95]) :

1. **Simulation de l'évolution continue.**

On résout symboliquement l'équation différentielle linéaire associée à l'état discret σ_d .

2. **Détection des événements.**

On calcule l'instant (si il existe) auquel la transition discrète survient. C'est l'instant t_f auquel la solution de l'équation différentielle sort du domaine D_{σ_d} .

3. Simulation de l'évolution discrète

On regarde quelles sont les transitions discrètes autorisées et on en choisit une.

La phase de simulation de l'évolution discrète est la seule qui puisse introduire de l'indéterminisme si plusieurs transitions sont autorisées à l'instant t_f . Ceci est en accord avec la proposition 3.2.2 du chapitre précédent.

Pour choisir parmi les transitions autorisées, on peut soit faire appel à un générateur aléatoire, soit donner une relation d'ordre sur les transitions.

<p>Données : Une condition initiale (t_0, σ_0, y_0) Un nombre maximal de transitions N.</p> <p>Résultat : Une exécution (τ, q, x).</p> <p>– <i>Initialisation</i> $\tau := \{\}; q := \{\}; x := \{\};$ $t_d := t_0; \sigma_d := \sigma_0; y_d := y_0;$ $i := 0;$</p> <p>tant que $t_d < +\infty$ <i>et</i> $i \leq N$ faire</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border-left: 1px solid black; padding-left: 10px; vertical-align: top;"> <p>– <i>Simulation de l'évolution continue</i> Résoudre symboliquement $y'(t) = A_{\sigma_d}y(t) + b_{\sigma_d}, y(t_d) = y_d;$</p> <p>si $y(t)$ <i>reste dans</i> D_{σ_d} <i>sur</i> $[t_d, +\infty[$ alors</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border-left: 1px solid black; padding-left: 10px;"> $\tau := \{\tau, [t_d, +\infty[; q := \{q, t \mapsto \sigma_d\}; x := \{x, t \mapsto y(t)\};$ $t_d := +\infty;$ </td> </tr> </table> <p>sinon</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border-left: 1px solid black; padding-left: 10px; vertical-align: top;"> <p>– <i>Détection des événements</i> Calculer t_f l'instant auquel $y(t)$ sort de D_{σ_d}; $\tau := \{\tau, [t_d, t_f]; q := \{q, t \mapsto \sigma_d\}; x := \{x, t \mapsto y(t)\};$</p> <p>– <i>Simulation de l'évolution discrète</i> Choisir une transition $e = (\sigma_d, \sigma_f)$ telle que $y(t_f) \in G_e$; $t_d := t_f; \sigma_d := \sigma_f; y_d := y(t_d);$ $i := i + 1;$</p> </td> </tr> </table> </td> <td style="border-left: 1px solid black; padding-left: 10px; vertical-align: top;"> <p>fin</p> </td> </tr> </table> <p>fin</p>	<p>– <i>Simulation de l'évolution continue</i> Résoudre symboliquement $y'(t) = A_{\sigma_d}y(t) + b_{\sigma_d}, y(t_d) = y_d;$</p> <p>si $y(t)$ <i>reste dans</i> D_{σ_d} <i>sur</i> $[t_d, +\infty[$ alors</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border-left: 1px solid black; padding-left: 10px;"> $\tau := \{\tau, [t_d, +\infty[; q := \{q, t \mapsto \sigma_d\}; x := \{x, t \mapsto y(t)\};$ $t_d := +\infty;$ </td> </tr> </table> <p>sinon</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border-left: 1px solid black; padding-left: 10px; vertical-align: top;"> <p>– <i>Détection des événements</i> Calculer t_f l'instant auquel $y(t)$ sort de D_{σ_d}; $\tau := \{\tau, [t_d, t_f]; q := \{q, t \mapsto \sigma_d\}; x := \{x, t \mapsto y(t)\};$</p> <p>– <i>Simulation de l'évolution discrète</i> Choisir une transition $e = (\sigma_d, \sigma_f)$ telle que $y(t_f) \in G_e$; $t_d := t_f; \sigma_d := \sigma_f; y_d := y(t_d);$ $i := i + 1;$</p> </td> </tr> </table>	$\tau := \{\tau, [t_d, +\infty[; q := \{q, t \mapsto \sigma_d\}; x := \{x, t \mapsto y(t)\};$ $t_d := +\infty;$	<p>– <i>Détection des événements</i> Calculer t_f l'instant auquel $y(t)$ sort de D_{σ_d}; $\tau := \{\tau, [t_d, t_f]; q := \{q, t \mapsto \sigma_d\}; x := \{x, t \mapsto y(t)\};$</p> <p>– <i>Simulation de l'évolution discrète</i> Choisir une transition $e = (\sigma_d, \sigma_f)$ telle que $y(t_f) \in G_e$; $t_d := t_f; \sigma_d := \sigma_f; y_d := y(t_d);$ $i := i + 1;$</p>	<p>fin</p>
<p>– <i>Simulation de l'évolution continue</i> Résoudre symboliquement $y'(t) = A_{\sigma_d}y(t) + b_{\sigma_d}, y(t_d) = y_d;$</p> <p>si $y(t)$ <i>reste dans</i> D_{σ_d} <i>sur</i> $[t_d, +\infty[$ alors</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border-left: 1px solid black; padding-left: 10px;"> $\tau := \{\tau, [t_d, +\infty[; q := \{q, t \mapsto \sigma_d\}; x := \{x, t \mapsto y(t)\};$ $t_d := +\infty;$ </td> </tr> </table> <p>sinon</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border-left: 1px solid black; padding-left: 10px; vertical-align: top;"> <p>– <i>Détection des événements</i> Calculer t_f l'instant auquel $y(t)$ sort de D_{σ_d}; $\tau := \{\tau, [t_d, t_f]; q := \{q, t \mapsto \sigma_d\}; x := \{x, t \mapsto y(t)\};$</p> <p>– <i>Simulation de l'évolution discrète</i> Choisir une transition $e = (\sigma_d, \sigma_f)$ telle que $y(t_f) \in G_e$; $t_d := t_f; \sigma_d := \sigma_f; y_d := y(t_d);$ $i := i + 1;$</p> </td> </tr> </table>	$\tau := \{\tau, [t_d, +\infty[; q := \{q, t \mapsto \sigma_d\}; x := \{x, t \mapsto y(t)\};$ $t_d := +\infty;$	<p>– <i>Détection des événements</i> Calculer t_f l'instant auquel $y(t)$ sort de D_{σ_d}; $\tau := \{\tau, [t_d, t_f]; q := \{q, t \mapsto \sigma_d\}; x := \{x, t \mapsto y(t)\};$</p> <p>– <i>Simulation de l'évolution discrète</i> Choisir une transition $e = (\sigma_d, \sigma_f)$ telle que $y(t_f) \in G_e$; $t_d := t_f; \sigma_d := \sigma_f; y_d := y(t_d);$ $i := i + 1;$</p>	<p>fin</p>	
$\tau := \{\tau, [t_d, +\infty[; q := \{q, t \mapsto \sigma_d\}; x := \{x, t \mapsto y(t)\};$ $t_d := +\infty;$				
<p>– <i>Détection des événements</i> Calculer t_f l'instant auquel $y(t)$ sort de D_{σ_d}; $\tau := \{\tau, [t_d, t_f]; q := \{q, t \mapsto \sigma_d\}; x := \{x, t \mapsto y(t)\};$</p> <p>– <i>Simulation de l'évolution discrète</i> Choisir une transition $e = (\sigma_d, \sigma_f)$ telle que $y(t_f) \in G_e$; $t_d := t_f; \sigma_d := \sigma_f; y_d := y(t_d);$ $i := i + 1;$</p>				

Algorithme 1: Simulation d'un système hybride

Dans les paragraphes suivants, nous détaillons les deux premières phases de chaque itération de la boucle. Pour alléger les notations, nous omettrons de noter les indices correspondants à l'état discret du système hybride. En effet, pendant une itération, la variable discrète $q(t)$ reste constante.

4.1 Simulation de l'évolution continue

La simulation de l'évolution continue consiste à résoudre un problème de Cauchy du type :

$$y'(t) = Ay(t) + b, \quad y(t_d) = y_d. \quad (4.1.1)$$

La solution de ce problème est bien connue et peut s'exprimer sous forme intégrale

$$y(t) = e^{A(t-t_d)}y_d + \int_{t_d}^t e^{A(t-s)}b \, ds.$$

Cependant, pour une évaluation numérique plus simple, il est souvent utile de procéder au changement de variable suivant :

$$z(t) = \begin{pmatrix} y(t) \\ 1 \end{pmatrix}, \quad y(t) = Pz(t).$$

La fonction z est alors solution de l'équation différentielle linéaire

$$z'(t) = \begin{pmatrix} y'(t) \\ 0 \end{pmatrix} = \begin{pmatrix} Ay(t) + b \\ 0 \end{pmatrix} = \begin{pmatrix} A & b \\ 0 & 0 \end{pmatrix} \begin{pmatrix} y(t) \\ 1 \end{pmatrix} = \begin{pmatrix} A & b \\ 0 & 0 \end{pmatrix} z(t).$$

En notant

$$\bar{A} = \begin{pmatrix} A & b \\ 0 & 0 \end{pmatrix},$$

on peut donner l'expression suivante de la fonction $y(t)$:

$$y(t) = Pe^{\bar{A}(t-t_d)} \begin{pmatrix} y_d \\ 1 \end{pmatrix}.$$

4.2 Détection des événements

La phase de détection des événements consiste à calculer l'instant t_f auquel survient la prochaine transition. Le problème peut être formulé de la manière suivante.

Soit y la solution du problème de Cauchy linéaire (4.1.1). t_f est le plus petit réel supérieur à t_d tel que $y(t_f)$ appartienne à l'ensemble des points de sortie du domaine polyédral D . Autrement dit, t_f est solution du problème

$$\begin{cases} \forall t \in [t_d, t_f], y(t) \in D, \\ \exists \varepsilon > 0, \forall t \in]t_f, t_f + \varepsilon[, y(t) \notin D. \end{cases} \quad (4.2.2)$$

Il est généralement impossible de résoudre ce problème de manière exacte. En effet, dès que l'on considère des équations différentielles de dimension supérieure ou égale à deux, il devient très difficile de trouver une expression formelle de la solution t_f [28].

On doit donc avoir recours à une résolution numérique du problème. Une méthode naïve consiste à calculer la valeur de $y(t)$ à certains points de discrétisation $\{t_d + kdt, k \in \mathbb{N}\}$. Dès que $y(t_d + kdt)$ n'est plus un élément du domaine, on en déduit que t_f est dans l'intervalle

$[t_d + (k - 1)dt, t_d + kdt]$. On peut alors affiner l'évaluation de t_f par une méthode de dichotomie.

Cependant, il y a certains cas où cette méthode échoue (voir figure 4.1). On voit que $y(t)$ quitte le domaine dans un intervalle $[t_d + (k - 1)dt, t_d + kdt]$ alors que les deux valeurs successives $y(t_d + (k - 1)dt)$ et $y(t_d + kdt)$ sont des éléments du domaine.

Ceci est problématique, à cause de la nature discontinue des systèmes hybrides. Si l'on ne détecte pas que $y(t)$ sort du domaine, la transition ne s'effectue pas et l'exécution calculée par l'algorithme de simulation peut être très différente de l'exécution réellement acceptée par le système hybride.

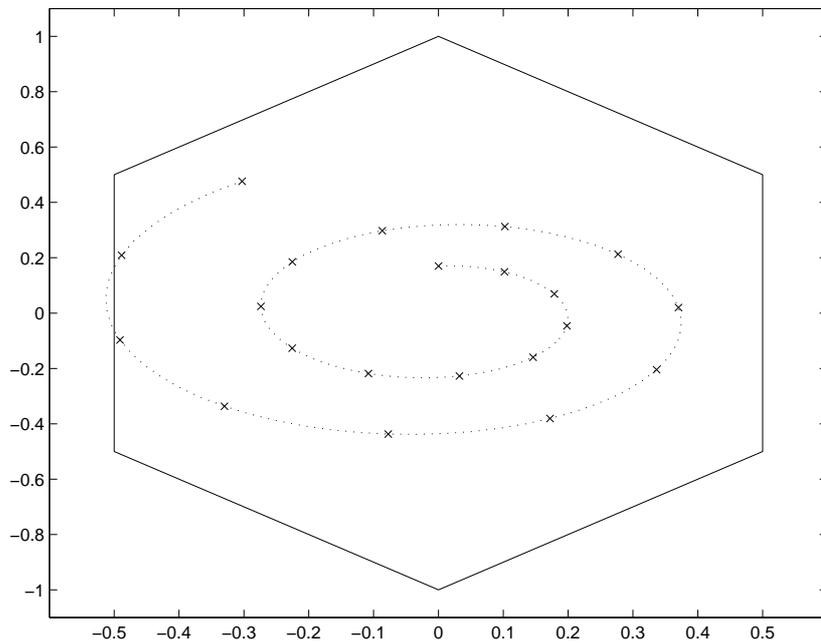


FIG. 4.1: Exemple de situation où la simulation naïve échoue

De nombreuses techniques ont été proposées pour résoudre le problème de détection des événements.

Dans [85], Shampine et al. proposent une méthode plus fiable pour tester si un événement survient durant l'intervalle $[t_d + (k - 1)dt, t_d + kdt]$. Elle consiste à construire une approximation polynomiale de la solution $y(t)$ sur l'intervalle et à vérifier si l'approximation sort du domaine. Park et Barton [76] ont combiné l'idée de l'approximation polynomiale avec des méthodes issues de l'arithmétique des intervalles pour déterminer les intervalles durant lesquels $y(t)$ peut sortir du domaine. Esposito et. al [38] ont utilisé des techniques de la théorie du contrôle pour calculer un pas de temps adaptatif, ralentissant la simulation quand on se rapproche de la frontière du domaine.

Toutes ces techniques sont basées sur des schémas d'approximation numérique des solutions des équations différentielles. Elles possèdent donc deux inconvénients majeurs. D'abord, la précision de la détection est évidemment limitée par celle du schéma numérique utilisé. De plus, le pas de temps doit rester suffisamment fin pour garder une précision acceptable, ce qui peut mener à un grand nombre d'itérations.

Dans les paragraphes qui suivent, nous présentons un algorithme (voir aussi [47]) pour la détection des événements dans les systèmes hybrides linéaires par morceaux. Il est basé sur l'expression symbolique de la solution $y(t)$ et, par conséquent, ne possède pas les défauts décrits précédemment.

4.2.1 Supprimer les contraintes superflues

A chaque facette du polyèdre D est associée une transition discrète du système hybride. E est l'ensemble de ces transitions. Ainsi le problème (4.2.2) peut être reformulé :

$$\begin{cases} \forall e \in E, \forall t \in [t_d, t_f], c_e^T y(t) \geq d_e, \\ \exists e \in E, \exists \varepsilon > 0, \forall t \in]t_f, t_f + \varepsilon[, c_e^T y(t) < d_e. \end{cases}$$

En notant $g_e(t) = c_e^T y(t) - d_e$, le problème à résoudre devient

$$\begin{cases} \forall e \in E, \forall t \in [t_d, t_f], g_e(t) \geq 0, \\ \exists e \in E, \exists \varepsilon > 0, \forall t \in]t_f, t_f + \varepsilon[, g_e(t) < 0. \end{cases} \quad (4.2.3)$$

Le problème peut encore être simplifié en supprimant certaines contraintes inutiles. Supposons que la condition initiale y_d est un élément de la frontière du polyèdre D . Il existe une transition $e \in E$, telle que $g_e(t_d) = c_e^T y_d - d_e = 0$. Trois configurations sont possibles :

Proposition 4.2.1 *Soit $e \in E$, telle que $g_e(t_d) = 0$,*

1. *Si $\forall k \in \{1, \dots, n\}$, $g_e^{(k)}(t_d) = c_e^T A^{k-1}(Ay_d + b) = 0$, alors $\forall t \in \mathbb{R}$, $g_e(t) = 0$.*
2. *Si $\exists p \in \{1, \dots, n\}$, $g_e^{(p)}(t_d) = c_e^T A^{p-1}(Ay_d + b) < 0$ et*

$$\forall k \in \{1, \dots, p-1\}, g_e^{(k)}(t_d) = c_e^T A^{k-1}(Ay_d + b) = 0,$$

alors il existe $\varepsilon > 0$, $\forall t \in]t_d, t_d + \varepsilon[, g_e(t) < 0$.

3. *Si $\exists p \in \{1, \dots, n\}$, $g_e^{(p)}(t_d) = c_e^T A^{p-1}(Ay_d + b) > 0$ et*

$$\forall k \in \{1, \dots, p-1\}, g_e^{(k)}(t_d) = c_e^T A^{k-1}(Ay_d + b) = 0,$$

alors il existe $\varepsilon > 0$, $\forall t \in]t_d, t_d + \varepsilon[, g_e(t) > 0$.

Preuve : Nous commençons par démontrer la première assertion.

A est une matrice carrée de taille $n \times n$, on peut donc exprimer A^n comme combinaison linéaire des matrices A^0, A, \dots, A^{n-1} : $A^n = \sum_{k=1}^{k=n} a_k A^{k-1}$. Par conséquent,

$$c_e^T A^n (Ay_d + b) = \sum_{k=1}^{k=n} a_k c_e^T A^{k-1} (Ay_d + b) = 0.$$

De proche en proche, on peut montrer que pour tout $k \in \mathbb{N}^*$, $c_e^T A^{k-1}(Ay_d + b) = 0$. En remarquant que, pour tout $t \in \mathbb{R}$, $y'(t) = e^{A(t-t_d)}(Ay_d + b)$, on arrive à

$$\forall t \in \mathbb{R}, g_e'(t) = c_e^T y'(t) = c_e^T e^{A(t-t_d)}(Ay_d + b) = \sum_{k=1}^{k=\infty} \frac{(t-t_d)^{k-1}}{(k-1)!} c_e^T A^{k-1}(Ay_d + b) = 0.$$

Donc $g_e(t)$ est constante sur \mathbb{R} et donc, pour tout $t \in \mathbb{R}$, $g_e(t) = 0$.

La deuxième et la troisième assertions sont évidentes.

■

Ainsi, si il existe une transition e vérifiant les hypothèses de la deuxième assertion, $y(t)$ quitte le polyèdre D à l'instant t_d , la transition du système hybride se produit alors à cet instant. Les transitions vérifiant les hypothèses de la première assertion peuvent être supprimées de l'ensemble E puisque l'on sait que les contraintes associées seront toujours respectées. On a alors, pour toutes les transitions $e \in E$ restantes :

$$\exists \varepsilon > 0, \forall t \in]t_d, t_d + \varepsilon[, g_e(t) > 0.$$

Le problème (4.2.3) peut donc se reformuler de la manière suivante :

$$\begin{cases} \forall e \in E, \forall t \in]t_d, t_f[, g_e(t) > 0, \\ \exists e \in E, g_e(t_f) = 0. \end{cases} \quad (4.2.4)$$

Remarque 4.2.1 *L'instant t_f , solution de ce problème, est le premier instant auquel $y(t)$ atteint la frontière du domaine D . Le point $y(t_f)$ n'est pas nécessairement un point de sortie. Donc t_f n'est pas toujours solution du problème (4.2.3). Grâce à un test, similaire à celui décrit plus haut, on peut déterminer si $y(t)$ sort de D à l'instant t_f . Sinon, on recommence en prenant t_f comme nouvel instant initial. Il convient cependant de noter que ce cas reste marginal et ne se produit presque jamais.*

Dans le paragraphe suivant, nous décrivons comment le problème ainsi transformé peut être résolu.

4.2.2 Algorithme de résolution

L'algorithme présenté dans cette partie utilise le fait qu'il est facile de calculer des sous-approximations et des sur-approximations de la solution du problème (4.2.4). Le détail de ces calculs sera exposé par la suite.

Données : L'ensemble de transitions E .

Résultat : Une sous-approximation \underline{t}_f et une sur-approximation \bar{t}_f de t_f .

répéter

 Calculer une sur-approximation \bar{t}_f de t_f ;

pour chaque $e \in E$ **faire**

 En supposant $g_e(t_f) = 0$,

 calculer une sous-approximation *hypothétique* $\underline{t}_f(e)$ de t_f ;

si $\underline{t}_f(e) > \bar{t}_f$ **alors**

 | $g_e(t_f) \neq 0$, supprimer e de E ;

fin

fin

$\underline{t}_f := \min_{e \in E} \underline{t}_f(e)$;

jusqu'à précision désirée atteinte;

Algorithme 2: Détection des événements

A chaque itération de l'algorithme, trois étapes principales doivent être effectuées.

1. D'abord, on calcule une sur-approximation \bar{t}_f de t_f .

2. Ensuite, on élimine de E les transitions dont on est sûr qu'elles ne se réaliseront pas. Pour chaque transition $e \in E$, on suppose que $g_e(t_f) = 0$. Cela nous permet de calculer une sous-approximation hypothétique $\underline{t}_f(e)$ de t_f .
On teste ensuite si $\underline{t}_f(e)$ est plus grande que \bar{t}_f . Si le test est positif, on arrive à une contradiction (la sous-approximation est plus grande que la sur-approximation). Cela signifie que l'hypothèse $g_e(t_f) = 0$ est fautive. Ainsi, on peut supprimer la transition e de E , sans changer le résultat du problème (4.2.4).
3. Enfin, le minimum de toutes les sous-approximations hypothétiques donne une sous-approximation effective de t_f . Pour nous en convaincre, raisonnons par l'absurde.
Si $\underline{t}_f > t_f$, alors, pour tout $e \in E$, $\underline{t}_f(e) > t_f$. Nécessairement, pour tout e , l'hypothèse $g_e(t_f) = 0$ est fautive. $y(t)$ ne sort donc jamais du domaine D et $t_f = +\infty$. Donc $\underline{t}_f \leq t_f$, ce qui amène la contradiction.

L'algorithme 2 permet de calculer des sous-approximations et des sur-approximations successives de t_f . Ces calculs seront détaillés par la suite, mais d'abord, il est nécessaire de montrer que les fonctions g_e vérifient certaines propriétés.

Propriétés des fonctions g_e

A partir de maintenant, on suppose que le domaine D est un polytope (un polyèdre borné).

Théorème 4.2.1 [101] *Si D est un polytope, alors il existe p points $\{y_1, \dots, y_p\}$ de \mathbb{R}^n tel que D est l'enveloppe convexe de ces points (voir figure 4.2).*

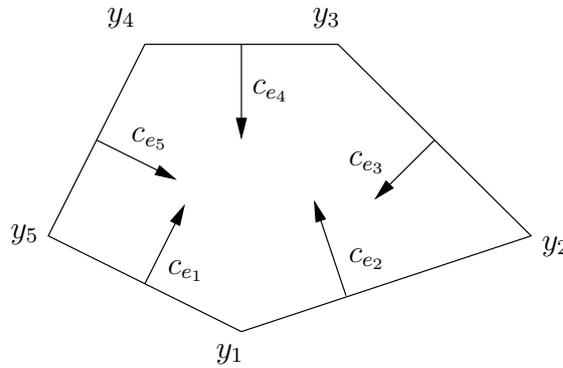


FIG. 4.2: Exemple d'un polytope D

On peut alors calculer des bornes pour les dérivées successives de g_e .

Proposition 4.2.2 *Soient $e \in E$, $k \geq 1$,*

$$\forall t \in [t_d, t_f], \begin{cases} g_e^{(k)}(t) \leq \max_{i=1}^{i=p} [c_e^T A^{k-1} (Ay_i + b)] = \bar{m}_e^k \\ g_e^{(k)}(t) \geq \min_{i=1}^{i=p} [c_e^T A^{k-1} (Ay_i + b)] = \underline{m}_e^k. \end{cases}$$

Preuve : Soient $e \in E$, $k \geq 1$, on a

$$g_e^{(k)}(t) = c_e^T y^{(k)}(t) = c_e^T A^{k-1} (Ay(t) + b).$$

Ensuite, pour tout $t \in [t_d, t_f]$, $y(t)$ est un élément du polytope D . Il existe donc p réels positifs $\{\lambda_1(t), \dots, \lambda_p(t)\}$ tels que

$$y(t) = \sum_{i=1}^{i=p} \lambda_i(t) y_i, \quad \sum_{i=1}^{i=p} \lambda_i(t) = 1.$$

En reportant dans l'expression de $g_e^{(k)}(t)$, on arrive au résultat énoncé. ■

On peut maintenant décrire le calcul des sur-approximations et des sous-approximations de t_f . Notons \underline{t}_f^j la valeur de la sous-approximation \underline{t}_f , \bar{t}_f^j la valeur de la sur-approximation \bar{t}_f après j itérations de l'algorithme 2. Ces suites sont initialisées de la manière suivante :

$$\underline{t}_f^0 = t_d, \quad \bar{t}_f^0 = +\infty.$$

Sur-approximation de t_f

Soit $e \in E$, il est clair que

$$\forall t \in [\underline{t}_f^j, t_f], \quad g_e(t) \leq g_e(\underline{t}_f^j) + (t - \underline{t}_f^j) g_e'(\underline{t}_f^j) + \frac{(t - \underline{t}_f^j)^2}{2} \overline{m}_e^2. \quad (4.2.5)$$

$g_e(t)$ est strictement positif sur l'intervalle $]\underline{t}_f^j, t_f[$. Donc, la plus petite racine strictement supérieure à \underline{t}_f^j de la sur-approximation polynomiale de $g_e(t)$ est, si elle existe, nécessairement plus grande que t_f . Ainsi, nous définissons la sur-approximation de t_f après $j + 1$ itérations par

$$\bar{t}_f^{j+1} = \min_{e \in E} (\bar{t}_f^{j+1}(e))$$

où $\bar{t}_f^{j+1}(e)$ est la plus petite racine strictement supérieure à \underline{t}_f^j de la sur-approximation polynomiale de $g_e(t)$ donnée par l'équation (4.2.5). Si une telle racine n'existe pas, on définit $\bar{t}_f^{j+1}(e) = +\infty$.

Sous-approximation de t_f

Nous avons vu que l'étape délicate de l'algorithme est le calcul d'une sous-approximation de t_f . Nous avons montré que cela était possible en utilisant un ensemble de sous-approximations hypothétiques. Dans ce paragraphe, nous détaillons le calcul de celles-ci.

Nous présentons d'abord le calcul de la première sous-approximation \underline{t}_f^1 de t_f .

Proposition 4.2.3 *Soit $e \in E$, il existe $p \in \{0, \dots, n\}$, tel que $g_e^{(p)}(t_d) > 0$*

$$\forall t \in [t_d, t_f], \quad g_e(t) \geq \frac{(t - t_d)^p}{p!} g_e^{(p)}(t_d) + \frac{(t - t_d)^{p+1}}{(p+1)!} \underline{m}_e^{p+1}.$$

Preuve : Si $g_e(t_d) \neq 0$, alors, en prenant $p = 0$, il est clair que la proposition est vraie. Si $g_e(t_d) = 0$, d'après la proposition 4.2.1, il existe $p \in \{1, \dots, n\}$ tel que, $g_e^{(p)}(t_d) > 0$ et

$$\forall k \in \{1, \dots, p-1\}, \quad g_e^{(k)}(t_d) = 0.$$

Or,

$$\forall t \in [t_d, t_f], g_e(t) \geq \sum_{k=0}^{k=p} \frac{(t-t_d)^k}{k!} g_e^{(k)}(t_d) + \frac{(t-t_d)^{p+1}}{(p+1)!} \underline{m}_e^{p+1}$$

ce qui permet de conclure. ■

Supposons $g_e(t_f) = 0$. En t_f , la sous-approximation polynomiale de $g_e(t)$, donnée par la proposition 4.2.3, est négative. On choisit donc $\underline{t}_f^1(e)$, de manière à ce que la sous-approximation de $g_e(t)$, et donc également $g_e(t)$, soient strictement positives sur $]t_d, \underline{t}_f^1(e)[$.

- Si $\underline{m}_e^{p+1} \geq 0$, $\underline{t}_f^1(e) = +\infty$.
- Si $\underline{m}_e^{p+1} < 0$, $\underline{t}_f^1(e) = t_d - g_e^{(p)}(t_d) \frac{p+1}{\underline{m}_e^{p+1}}$.

La première sous-approximation de t_f est alors donnée par

$$\underline{t}_f^1 = \min_{e \in E} \underline{t}_f^1(e).$$

Remarque 4.2.2 *Il est important de noter que $t_d < \underline{t}_f^1 \leq t_f$, par conséquent $g_e(\underline{t}_f^1) = 0$ si et seulement si $\underline{t}_f^1 = t_f$.*

La suite des sous-approximations est croissante, donc, pour tout $j \geq 1$, $\underline{t}_f^1 \leq \underline{t}_f^j \leq t_f$. Ainsi, $g_e(\underline{t}_f^j) = 0$ si et seulement si $\underline{t}_f^j = t_f$.

Nous présentons maintenant le calcul des sous-approximations suivantes. Soit $e \in \mathcal{E}$, nous avons

$$\forall t \in [\underline{t}_f^j, t_f], g_e(t) \geq g_e(\underline{t}_f^j) + (t - \underline{t}_f^j) g_e'(\underline{t}_f^j) + \frac{(t - \underline{t}_f^j)^2}{2} \underline{m}_e^2. \quad (4.2.6)$$

Supposons $g_e(t_f) = 0$. En t_f , la sous-approximation polynomiale de $g_e(t)$, donnée par l'équation (4.2.6), est négative. Soit $\underline{t}_f^{j+1}(e)$ la plus petite racine, strictement supérieure à \underline{t}_f^j , de la sous-approximation de $g_e(t)$. Ce polynôme est strictement positif en \underline{t}_f^j et donc également sur $[\underline{t}_f^j, \underline{t}_f^{j+1}(e)[$. t_f est donc nécessairement plus grand que $\underline{t}_f^{j+1}(e)$.

Comme nous l'avons expliqué précédemment (voir l'algorithme 2), on prend ensuite

$$\underline{t}_f^{j+1} = \min_{e \in E} (\underline{t}_f^{j+1}(e)).$$

Remarque 4.2.3 *Nous sommes obligés de différencier le calcul de \underline{t}_f^1 et celui des sous-approximations suivantes. En effet, supposons qu'il existe une transition $e \in E$ telle que $g_e(t_d) = g_e'(t_d) = 0$ et $\underline{m}_e^2 < 0$. Si nous n'appliquons pas le calcul spécifique de \underline{t}_f^1 alors nous avons $\underline{t}_f^1 = t_d$ et pour tout $j \in \mathbb{N}$, $\underline{t}_f^j = t_d$.*

Il faut cependant se rendre compte que ce cas est très particulier. En pratique, il est souvent inutile d'appliquer le calcul spécifique de \underline{t}_f^1 .

4.2.3 Convergence et tests

Dans cette partie, nous montrons que les suites (\underline{t}_f^j) et (\bar{t}_f^j) donnent un encadrement convergent de t_f .

Théorème 4.2.2 (Convergence des suites (\underline{t}_f^j) et (\bar{t}_f^j))

$$\lim_{j \rightarrow +\infty} \underline{t}_f^j = t_f.$$

Si $y(t)$ n'atteint pas la frontière de D tangentiellement,

$$\lim_{j \rightarrow +\infty} \bar{t}_f^j = t_f.$$

Si de plus, il existe une unique transition $e \in E$ telle que $g_e(t_f) = 0$, alors

$$|\bar{t}_f^{j+1} - \underline{t}_f^{j+1}| = O(|\bar{t}_f^j - \underline{t}_f^j|^2).$$

Preuve : Notons que, par construction, la suite (\underline{t}_f^j) est croissante et majorée par t_f . Elle est donc convergente et sa limite \underline{t}_f^∞ est dans l'intervalle $]t_d, t_f]$. De plus, \underline{t}_f^∞ doit être un point fixe de l'itération définissant \underline{t}_f^{j+1} à partir de \underline{t}_f^j . Donc, il existe une transition $e \in E$ tel que $g_e(\underline{t}_f^\infty) = 0$. Nécessairement, $\underline{t}_f^\infty = t_f$.

Supposons maintenant que $y(t)$ n'atteint pas la frontière de D tangentiellement. On a $g_e(t_f) = 0$ et $g'_e(t_f) < 0$. Par continuité des applications g_e et g'_e , on a également,

$$\lim_{j \rightarrow +\infty} g_e(\underline{t}_f^j) = 0 \text{ et } \lim_{j \rightarrow +\infty} g'_e(\underline{t}_f^j) < 0$$

Il existe donc $J \in \mathbb{N}$, tel que, pour tout $j \geq J$, le discriminant de la sur-approximation polynomiale donnée par l'équation (4.2.5) soit strictement positif :

$$\forall j \geq J, g'_e(\underline{t}_f^j)^2 - 2\bar{m}_e^2 g_e(\underline{t}_f^j) > 0.$$

On peut alors calculer la valeur de la sur-approximation $\bar{t}_f^{j+1}(e)$:

– Si $\bar{m}_e^2 = 0$, alors

$$\forall j \geq J, \bar{t}_f^{j+1}(e) = \underline{t}_f^j + \frac{-g_e(\underline{t}_f^j)}{g'_e(\underline{t}_f^j)}.$$

– Si $\bar{m}_e^2 \neq 0$, alors

$$\forall j \geq J, \bar{t}_f^{j+1}(e) = \underline{t}_f^j + \frac{-g'_e(\underline{t}_f^j) - \sqrt{g'_e(\underline{t}_f^j)^2 - 2\bar{m}_e^2 g_e(\underline{t}_f^j)}}{\bar{m}_e^2}.$$

Dans les deux cas, $\bar{t}_f^{j+1}(e)$ tend vers t_f quand j tend vers l'infini. De plus,

$$t_f \leq \bar{t}_f^{j+1} \leq \bar{t}_f^{j+1}(e).$$

Donc \bar{t}_f^{j+1} tend vers t_f quand j tend vers l'infini.

Calculons maintenant l'ordre de convergence. Si il existe une unique transition $e \in E$ telle que $g_e(t_f) = 0$, il est clair qu'à partir d'un certain rang $\underline{t}_f^j = \underline{t}_f^j(e)$ et $\bar{t}_f^j = \bar{t}_f^j(e)$. On a alors

$$\begin{cases} g_e(\underline{t}_f^j) + (\underline{t}_f^{j+1} - \underline{t}_f^j)g'_e(\underline{t}_f^j) + \frac{(\underline{t}_f^{j+1} - \underline{t}_f^j)^2}{2}m_e^2 = 0 \\ g_e(\bar{t}_f^j) + (\bar{t}_f^{j+1} - \bar{t}_f^j)g'_e(\bar{t}_f^j) + \frac{(\bar{t}_f^{j+1} - \bar{t}_f^j)^2}{2}\bar{m}_e^2 = 0 \end{cases}$$

D'où,

$$(\bar{t}_f^{j+1} - \underline{t}_f^{j+1})g'_e(\underline{t}_f^j) = \frac{(\underline{t}_f^{j+1} - \underline{t}_f^j)^2}{2}m_e^2 - \frac{(\bar{t}_f^{j+1} - \bar{t}_f^j)^2}{2}\bar{m}_e^2.$$

Par conséquent,

$$|\bar{t}_f^{j+1} - \underline{t}_f^{j+1}||g'_e(\underline{t}_f^j)| \leq \frac{|m_e^2|}{2}|\underline{t}_f^{j+1} - \underline{t}_f^j|^2 + \frac{|\bar{m}_e^2|}{2}|\bar{t}_f^{j+1} - \bar{t}_f^j|^2.$$

En remarquant que $|\underline{t}_f^{j+1} - \underline{t}_f^j| \leq |\bar{t}_f^j - \underline{t}_f^j|$ (car $\underline{t}_f^j \leq \underline{t}_f^{j+1} \leq t_f \leq \bar{t}_f^j$) et $|\bar{t}_f^{j+1} - \bar{t}_f^j| \leq |\bar{t}_f^j - \underline{t}_f^j|$ (car $\underline{t}_f^j \leq t_f \leq \bar{t}_f^{j+1} \leq \bar{t}_f^j$), on arrive au résultat souhaité. ■

Nous avons appliqué l'algorithme de détection des événements sur l'exemple suivant (voir aussi figure 4.1).

$$\begin{cases} y_1'(t) = 0.1y_1(t) + y_2(t) & , y_1(0) = 0 \\ y_2'(t) = -y_1(t) + 0.1y_2(t) & , y_2(0) = 0.17. \end{cases}$$

Nous définissons le polytope D comme l'enveloppe convexe de l'ensemble de points

$$\{(0, 1), (-0.5, 0.5), (-0.5, -0.5), (0, -1), (0.5, -0.5), (0.5, 0.5)\}.$$

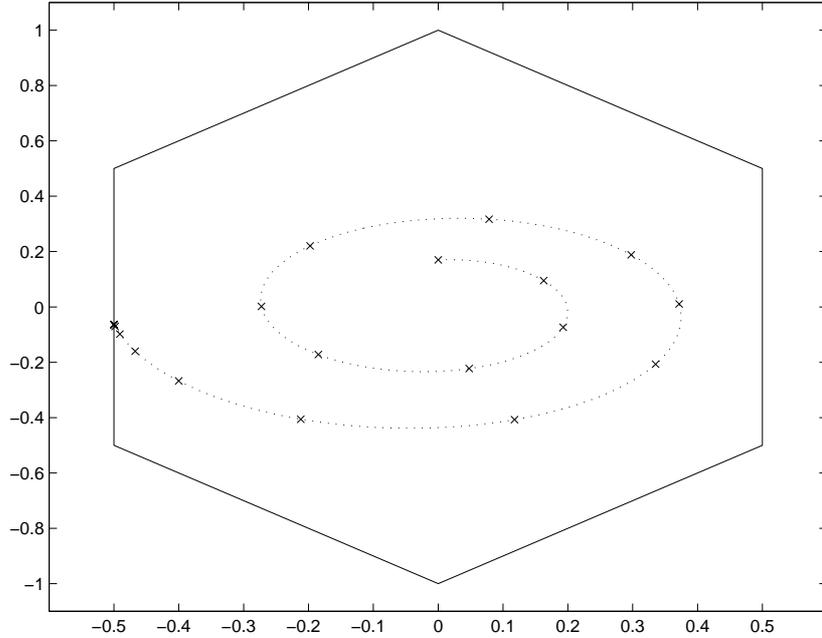


FIG. 4.3: Calculs successifs de $y(\underline{t}_f^j)$

La valeur approximative de t_f calculée par notre algorithme est 10.8687. Sur la figure 4.3, nous avons représenté la valeur de $y(t)$ aux instants \underline{t}_f^j . On voit que notre algorithme permet d'utiliser une discrétisation relativement large quand on est loin des facettes de D . Expérimentalement, on retrouve bien que les suites (\underline{t}_f^j) et (\bar{t}_f^j) convergent de manière quadratique vers t_f .

Chapitre 5

Exécutions Périodiques d'un Système Hybride

Les systèmes hybrides exhibent fréquemment des comportements périodiques (voir par exemple [60, 80]). Par conséquent, il est nécessaire de développer des outils mathématiques pour le calcul et l'analyse de la stabilité des exécutions périodiques.

Les travaux menés sur ce sujet ont d'abord porté sur la généralisation du théorème de Poincaré Bendixon et l'analyse de stabilité.

L'extension du théorème de Poincaré Bendixon s'applique aux systèmes hybrides plans (la variable continue prend ses valeurs dans \mathbb{R}^2).

Le théorème de Poincaré Bendixon est un résultat classique de la théorie qualitative des équations différentielles (voir par exemple [78]). Il permet de caractériser le bassin d'attraction des orbites périodiques des systèmes dynamiques plans.

Ainsi, Matveev et Savkin [73] ont généralisé le théorème de Poincaré Bendixon aux systèmes hybrides constants par morceaux plans. Asarin, Schneider et Yovine ont étendu ce résultat aux inclusions différentielles constantes par morceaux [5]. Enfin, pour les systèmes hybrides déterministes plans, Simic, Johansson, Lygeros et Sastry ont démontré un théorème de Poincaré Bendixon hybride [88].

Pour l'analyse de stabilité, les méthodes développées utilisent soit des résultats de la théorie de la stabilité au sens de Lyapunov [52, 81], soit la notion d'analyse de sensibilité des trajectoires (sensibilité par rapport aux paramètres et aux conditions initiales) [61, 59].

Cependant, le problème du calcul des exécutions périodiques a été peu traité. Dans [69], des équations symboliques, non-triviales, ont été établies pour le calcul des cycles limites dans un système dynamique plan, continu et linéaire par morceaux avec 2 régions. Même dans ce cas simple, l'utilisation de méthodes numériques est nécessaire pour résoudre le système de deux équations à deux inconnues.

Ainsi, il semble indispensable de développer des méthodes numériques pour calculer les exécutions périodiques acceptées par un système hybride linéaire par morceaux quelconque et donc nécessairement plus complexe.

Dans [55, 97], des algorithmes numériques efficaces ont été proposés pour le calcul des solutions périodiques dans les systèmes dynamiques continus.

Dans ce chapitre, nous proposons d'étendre ces techniques au calcul des exécutions périodiques des systèmes hybrides linéaires par morceaux. Notre méthode (voir aussi [48]) repose sur la généralisation du concept d'application de Poincaré aux systèmes hybrides.

5.1 Définition, exemple

La notion d'exécution périodique est assez intuitive. Cela signifie qu'il existe une période T , strictement positive, telle que les variables (discrète et continue) du système hybride ont la même valeur à un instant $t \in \tau$ et à l'instant $t + T$.

De proche en proche, on montre que les variables q et x ont la même valeur à l'instant $t + nT$, pour $n \in \mathbb{N}$. Par conséquent, la trajectoire temporisée associée à une exécution périodique est nécessairement illimitée.

Définition 5.1.1 (Exécution périodique) Une exécution (τ, q, x) est dite périodique si :

- la trajectoire temporisée τ est illimitée;
- il existe $T > 0$ tel que

$$\forall t \in \tau, q(t + T) = q(t), x(t + T) = x(t).$$

T est la période de l'exécution.

Illustrons cette définition grâce à un exemple.

Exemple 5.1.1 (Oscillateur électronique à valve [1, 79]) Reprenons l'exemple 1.2.2 présenté dans le premier chapitre. On suppose que l'intensité de saturation est égale à 1, ($I_s = 1$). En introduisant les constantes

$$\omega_0^2 = \frac{1}{LC} \text{ et } 2h = \frac{R}{L},$$

les équations régissant le système deviennent

$$x_1'(t) = x_2(t), x_2'(t) = \begin{cases} -\omega_0^2 x_1(t) - 2hx_2(t) & \text{si } x_2(t) < 0, \\ -\omega_0^2 x_1(t) - 2hx_2(t) + \omega_0^2 & \text{si } x_2(t) > 0. \end{cases}$$

On suppose que $\omega^2 = \omega_0^2 - h^2 > 0$, ainsi le circuit RLC est un oscillateur amorti.

A l'instant initial $t_0 = 0$, le système n'est pas saturé ($q(0) = 1$), on impose la condition initiale $x_1(0) = i_0 > 0$, $x_2(0) = 0$. Tant que $x_2(t) \leq 0$, on a

$$\begin{cases} x_1(t) = i_0 e^{-ht} \left(\cos(\omega t) + \frac{h}{\omega} \sin(\omega t) \right) \\ x_2(t) = -i_0 \frac{\omega^2 + h^2}{\omega} e^{-ht} \sin(\omega t). \end{cases}$$

A l'instant $t_1 = \frac{\pi}{\omega}$, x_2 s'annule à nouveau, le système hybride effectue alors une transition vers l'état de saturation ($q(t_1^+) = 2$). A l'instant t_1 ,

$$x_1(t_1) = -i_0 e^{-\frac{\pi h}{\omega}}.$$

On a alors, tant que $x_2(t) \geq 0$,

$$\begin{cases} x_1(t) = 1 - (i_0 e^{-\frac{\pi h}{\omega}} + 1) e^{-h(t-t_1)} \left(\cos(\omega(t-t_1)) + \frac{h}{\omega} \sin(\omega(t-t_1)) \right) \\ x_2(t) = (i_0 e^{-\frac{\pi h}{\omega}} + 1) \frac{\omega^2 + h^2}{\omega} e^{-h(t-t_1)} \sin(\omega(t-t_1)). \end{cases}$$

A l'instant $t_2 = \frac{2\pi}{\omega}$, x_2 s'annule à nouveau, le système hybride effectue alors une transition vers l'état non saturé ($q(t_2^+) = 1$). A l'instant t_2 ,

$$x_1(t_2) = 1 + e^{-\frac{\pi h}{\omega}} + e^{-\frac{2\pi h}{\omega}} i_0.$$

Remarquons que si

$$i_0 = \frac{1 + e^{-\frac{\pi h}{\omega}}}{1 - e^{-\frac{2\pi h}{\omega}}} = \frac{1}{1 - e^{-\frac{\pi h}{\omega}}},$$

alors $x_1(t_0) = x_1(t_2)$. Ainsi, pour la condition initiale $(0, 1, (i_0, 0))$, le système hybride accepte une exécution périodique de période $\frac{2\pi}{\omega}$.

Soit (τ, q, x) une exécution périodique acceptée par un système hybride \mathcal{H} . On a vu que τ est nécessairement illimitée.

Si τ est finie, il y a un nombre fini de transitions. D'après l'hypothèse de périodicité de la variable discrète q , on peut conclure que le système hybride n'effectue aucune transition :

$$\forall t \in \tau, q(t) = \sigma.$$

Dans ce cas, l'existence de l'exécution périodique correspond à l'existence d'une solution périodique de l'équation différentielle associée au champ de vecteurs f_σ .

Dans le cas d'un système hybride linéaire par morceaux, cela signifie donc que la matrice A_σ possède des valeurs propres imaginaires pures. Par la suite, nous ne nous intéresserons pas à ce cas. En effet, il peut être traité par les méthodes classiques pour les systèmes continus.

Si τ est infinie, la trajectoire Γ parcourue dans \mathbb{R}^n sur une période par la variable continue est composée de morceaux de solutions d'équations différentielles linéaires. Dans ce cas, Γ est appelée cycle hybride.

Il existe alors un entier N , des nombres réels $0 = t_0^* \leq t_1^* \leq \dots \leq t_{N+1}^* = T$ et une séquence d'états discrets $\{\sigma_0, \dots, \sigma_N\}$ tels que le système hybride accepte une exécution (τ^*, q^*, x^*) vérifiant :

$$\tau^* = \{[t_i^*, t_{i+1}^*]\}_{i=0}^{i=N}, \quad q^* = \{t \rightarrow \sigma_i\}_{i=0}^{i=N}$$

et

$$x^*(0) = x^*(t_{N+1}^*) \in G_{(\sigma_N, \sigma_0)}.$$

L'exécution (τ^*, q^*, x^*) est appelée motif de l'exécution périodique. En effet, l'exécution périodique acceptée par le système hybride est formée d'une infinité de répétitions du motif (τ^*, q^*, x^*) .

Durant une période T le système hybride effectue donc N transitions :

$$\begin{cases} e_i &= (\sigma_i, \sigma_{i+1}), \quad i \in \{0, \dots, N-1\} \\ e_N &= (\sigma_N, \sigma_0). \end{cases}$$

On définit également les grandeurs suivantes :

$$\begin{cases} y_i^* &= x^*(t_i^*), \quad i \in \{0, \dots, N+1\} \\ s_i^* &= t_{i+1}^* - t_i^*, \quad i \in \{0, \dots, N\}. \end{cases}$$

Notons que, $y_0^* = y_{N+1}^*$ et que pour tout $i \in \{0, \dots, N\}$, $y_{i+1}^* \in G_{e_i}$.

On fera également les hypothèses suivantes (voir figure 5.1). Pour tout $i \in \{0, \dots, N\}$:

1. Il existe $\eta > 0$, tel que, pour tout $e \in E_{\sigma_i}$, $e \neq e_i$,

$$\forall t \in [t_i^*, t_{i+1}^*], \mathcal{B}(x^*(t), \eta) \cap G_e = \emptyset,$$

où $\mathcal{B}(x^*(t), \eta)$ est la boule centrée en $x^*(t)$ de rayon η .

2. $x^*(t)$ n'atteint pas la garde G_{e_i} tangentiellement :

$$c_{e_i}^T (A_{\sigma_i} y_{i+1}^* + b_{\sigma_i}) \neq 0.$$

3. Pour tout $\varepsilon > 0$, il existe $\delta > 0$ tel que

$$\forall t \in [t_i^*, t_{i+1}^* - \varepsilon], \mathcal{B}(x^*(t), \delta) \cap G_{e_i} = \emptyset.$$

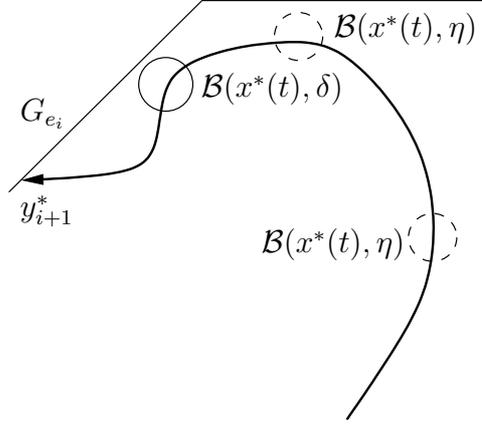


FIG. 5.1: Illustration des hypothèses sur le cycle hybride

Ces hypothèses sont liées à la stabilité de l'exécution périodique. Elles garantissent que, pour des conditions initiales $(0, \sigma_0, x_0)$ avec x_0 dans un voisinage de x_0^* , les $N + 1$ premiers éléments de la séquence des modes discrets de l'exécution acceptée par le système hybride \mathcal{H} , sont les mêmes que ceux du cycle.

La première hypothèse signifie, qu'entre deux transitions du système hybride, la variable continue $x^*(t)$ est à une distance minimale de η des gardes des autres transitions. Ainsi, on peut perturber légèrement la trajectoire $x^*(t)$ sans que celle-ci n'atteigne une de ces gardes. La seule transition possible est donc e_i .

Si on ne fait pas la deuxième hypothèse, alors une légère perturbation de la trajectoire peut suffire à ce que $x^*(t)$ n'atteigne plus G_{e_i} . La transition ne peut alors pas se faire.

La troisième hypothèse garantit que la transition e_i ne peut se faire qu'à un instant voisin de t_{i+1}^* .

5.2 Application de Poincaré hybride

Dans cette partie, nous étendons la notion d'application de Poincaré aux systèmes hybrides. Mais d'abord, il nous semble utile d'introduire cet outil dans le cadre des systèmes dynamiques (on pourra aussi voir [62]).

5.2.1 Application de Poincaré pour les systèmes dynamiques

Le calcul des solutions périodiques est un problème important de la théorie des systèmes dynamiques. Un des outils les plus efficaces pour traiter ce problème est l'application de Poincaré. Le principe est illustré sur la figure 5.2.

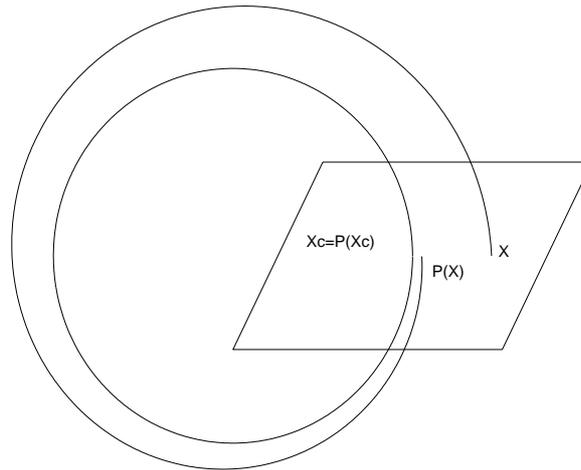


FIG. 5.2: Application de Poincaré pour un système dynamique

Considérons un système dynamique avec une solution périodique. Dans l'espace des phases (disons \mathbb{R}^n), ceci se traduit par l'existence d'un cycle Γ .

Soit \mathcal{S} un hyperplan transverse au cycle Γ en un point x_c . On définit l'application de Poincaré P sur un voisinage de x_c . A tout point x de l'hyperplan \mathcal{S} , on associe le point $P(x)$, tel que la trajectoire du système dynamique, issue de x , revient sur l'hyperplan \mathcal{S} au point $P(x)$ après approximativement une période de Γ .

Calculer la solution périodique devient alors équivalent à trouver le point fixe de l'application de Poincaré. De plus, si l'on est capable de calculer la différentielle de cette application, alors la méthode de Newton appliquée à l'application P fournit une méthode efficace pour le calcul du cycle limite [55, 97].

Dans les paragraphes suivants, nous montrons que cette notion peut être généralisée aux systèmes hybrides linéaires par morceaux.

5.2.2 Application d'impact

Les applications d'impact sont des outils relativement répandus pour l'analyse des systèmes hybrides [20, 52, 95].

Soit $\sigma \in \mathcal{Q}$, l'application d'impact P_σ est celle qui à un point y du domaine D_σ , associe le point, si il existe, par lequel la solution de l'équation différentielle associée à l'état σ passant par y sort du domaine D_σ . P_σ est donc une application d'une partie de D_σ dans S_σ l'ensemble des points de sortie du domaine D_σ . La dynamique continue du système est ainsi *abstraite*, la notion de temps ayant été supprimée.

Etant donnée une condition initiale (t_0, σ_0, y_0) , on sait ainsi qu'un événement discret surviendra, mais on ignore à quel instant, lorsque la variable continue vaudra $P_{\sigma_0}(y_0)$. Une transition $e_0 = (\sigma_0, \sigma_1)$, telle que $P_{\sigma_0}(y_0) \in G_{e_0}$, s'effectue alors. On considère ensuite l'application P_{σ_1} . De proche en proche, on peut déterminer la séquence des états discrets et la valeur de la variable continue lors des transitions.

On s'intéresse à ses applications au voisinage du cycle hybride (voir figure 5.3). Tout d'abord, notons que pour tout $i \in \{0, \dots, N\}$, $y_{i+1}^* = P_{\sigma_i}(y_i^*)$. Au voisinage de y_i^* , on peut montrer que l'application P_{σ_i} vérifie certaines propriétés.

Théorème 5.2.1 *Pour tout $i \in \{0, \dots, N\}$, il existe un voisinage $V(y_i^*)$ de y_i^* tel que P_{σ_i} est définie sur $V(y_i^*)$, et*

$$\forall y_i \in V(y_i^*), P_{\sigma_i}(y_i) \in G_{e_i}.$$

On note $y_{i+1} = P_{\sigma_i}(y_i)$.

L'application d'impact P_{σ_i} est de classe C^∞ sur $V(y_i^*)$ et sa différentielle en y_i est

$$dP_{\sigma_i}(y_i) = \left(I - \frac{(A_{\sigma_i} y_{i+1} + b_{\sigma_i}) c_{e_i}^T}{c_{e_i}^T (A_{\sigma_i} y_{i+1} + b_{\sigma_i})} \right) e^{s_i A_{\sigma_i}}$$

où s_i est le temps que met la solution de l'équation différentielle linéaire associée à l'état σ_i pour aller de y_i à y_{i+1} .

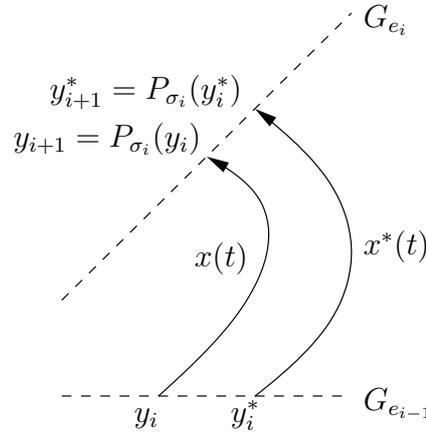


FIG. 5.3: Application d'impact P_{σ_i}

Preuve : Soit Φ_{σ_i} le flot de l'équation différentielle associée à l'état σ_i :

$$\Phi_{\sigma_i}(y, s) = e^{A_{\sigma_i} s} y + \int_0^s e^{A_{\sigma_i}(s-\tau)} b_{\sigma_i} d\tau.$$

Φ_{σ_i} est de classe C^∞ par rapport aux variables y et s .

Remarquons que

$$\forall s \in [0, s_i^*], \Phi_{\sigma_i}(y_i^*, s) = x^*(t_i^* + s).$$

Montrons que pour y_i voisin de y_i^* , la trajectoire issue de y_i n'atteint pas la garde G_e (pour $e \neq e_i$).

Soit $\eta > 0$, la constante donnée par l'hypothèse 1. Par continuité de Φ_{σ_i} par rapport à s , il existe $\bar{s}_i > s_i^*$, tel que

$$\forall s \in [s_i^*, \bar{s}_i], \|\Phi_{\sigma_i}(y_i^*, s) - \Phi_{\sigma_i}(y_i^*, s_i^*)\| < \frac{\eta}{2}.$$

De plus, il existe $V_1(y_i^*)$, un voisinage de y_i^* , tel que

$$\forall y_i \in V_1(y_i^*), \forall s \in [0, \bar{s}_i], \|\Phi_{\sigma_i}(y_i, s) - \Phi_{\sigma_i}(y_i^*, s)\| \leq e^{\|A_{\sigma_i} \bar{s}_i\|} \|y_i - y_i^*\| < \frac{\eta}{2}. \quad (5.2.1)$$

Donc, pour tout $y_i \in V_1(y_i^*)$ et pour tout $s \in [s_i^*, \bar{s}_i]$,

$$\|\Phi_{\sigma_i}(y_i, s) - \Phi_{\sigma_i}(y_i^*, s_i^*)\| \leq \|\Phi_{\sigma_i}(y_i, s) - \Phi_{\sigma_i}(y_i^*, s)\| + \|\Phi_{\sigma_i}(y_i^*, s) - \Phi_{\sigma_i}(y_i^*, s_i^*)\| < \eta. \quad (5.2.2)$$

On peut donc conclure, grâce à l'hypothèse 1 et aux équations (5.2.1) et (5.2.2) que

$$\forall y_i \in V_1(y_i^*), \forall s \in [0, \bar{s}_i], \forall e \in E_{\sigma_i}, e \neq e_i, \Phi_{\sigma_i}(y_i, s) \notin G_e. \quad (5.2.3)$$

Soit y_i dans un voisinage de y_i^* , on désire calculer l'instant s_i tel que $\Phi_{\sigma_i}(y_i, s_i) \in G_{e_i}$:

$$H_{\sigma_i}(y_i, s_i) = c_{e_i}^T \Phi_{\sigma_i}(y_i, s_i) - d_{e_i} = 0.$$

L'application H_{σ_i} est C^∞ par rapport aux deux variables. De plus, $H_{\sigma_i}(y_i^*, s_i^*) = 0$ et d'après l'hypothèse 2 :

$$\frac{\partial H_{\sigma_i}}{\partial s}(y_i^*, s_i^*) = c_{e_i}^T \frac{\partial \Phi_{\sigma_i}}{\partial s}(y_i^*, s_i^*) = c_{e_i}^T (A_{\sigma_i} y_{i+1}^* + b_{\sigma_i}) \neq 0.$$

Le théorème des fonctions implicites s'applique, il existe :

- $I = [\underline{s}_i, \bar{s}_i']$, un intervalle contenant s_i^* , avec $\underline{s}_i < s_i^* < \bar{s}_i' \leq \bar{s}_i$;
- $V_2(y_i^*)$, un voisinage de y_i^* ;
- s_{σ_i} , une application de classe C^∞ de $V_2(y_i^*)$ dans I

tels que

$$\forall (y_i, s_i) \in V_2(y_i^*) \times I, H_{\sigma_i}(y_i, s_i) = 0 \iff s_i = s_{\sigma_i}(y_i). \quad (5.2.4)$$

D'après l'hypothèse 3,

$$\exists \delta, \forall s \in [0, \underline{s}_i], \mathcal{B}(\Phi_{\sigma_i}(y_i^*, s), \delta) \cap G_{e_i} = \emptyset.$$

Il existe $V_3(y_i^*)$, un voisinage de y_i^* tel que

$$\forall y_i \in V_3(y_i^*), \forall s \in [0, \underline{s}_i], \|\Phi_{\sigma_i}(y_i, s) - \Phi_{\sigma_i}(y_i^*, s)\| \leq e^{\|A_{\sigma_i}\| s_i^*} \|y_i - y_i^*\| < \delta.$$

Donc,

$$\forall y_i \in V_3(y_i^*), \forall s \in [0, \underline{s}_i], \Phi_{\sigma_i}(y_i, s) \notin G_{e_i}. \quad (5.2.5)$$

Soit $V(y_i^*) = V_1(y_i^*) \cap V_2(y_i^*) \cap V_3(y_i^*)$. D'après (5.2.3), (5.2.4) et (5.2.5), pour tout $y_i \in V(y_i^*)$:

- $\forall s \in [0, s_{\sigma_i}(y_i)], \forall e \in E_{\sigma_i}, e \neq e_i, \Phi_{\sigma_i}(y_i, s) \notin G_e$;
- $\forall s \in [0, s_{\sigma_i}(y_i)[, \Phi_{\sigma_i}(y_i, s) \notin G_{e_i}$;
- $\Phi_{\sigma_i}(y_i, s_{\sigma_i}(y_i)) \in G_{e_i}$.

Donc l'application d'impact, P_{σ_i} , est définie sur $V(y_i^*)$ et

$$\forall y_i \in V(y_i^*), P_{\sigma_i}(y_i) = \Phi_{\sigma_i}(y_i, s_{\sigma_i}(y_i)) \in G_{e_i}.$$

De plus P_{σ_i} est de classe C^∞ et sa différentielle est donnée par

$$\begin{aligned} dP_{\sigma_i}(y_i) &= \frac{\partial \Phi_{\sigma_i}(y_i, s_i)}{\partial y} + \frac{\partial \Phi_{\sigma_i}(y_i, s_i)}{\partial s} ds_{\sigma_i}(y_i) \\ &= e^{A_{\sigma_i} s_{\sigma_i}(y_i)} + (A_{\sigma_i} \Phi_{\sigma_i}(y_i, s_i) + b_{\sigma_i}) ds_{\sigma_i}(y_i). \end{aligned}$$

Or d'après le théorème des fonctions implicites :

$$ds_{\sigma_i}(y_i) = -\frac{\frac{\partial H_{\sigma_i}}{\partial y}(y_i, s_{\sigma_i}(y_i))}{\frac{\partial H_{\sigma_i}}{\partial s}(y_i, s_{\sigma_i}(y_i))} = \frac{-c_{e_i}^T e^{A_{\sigma_i} s_{\sigma_i}(y_i)}}{c_{e_i}^T (A_{\sigma_i} \Phi_{\sigma_i}(y_i, s_{\sigma_i}(y_i)) + b_{\sigma_i})}.$$

En notant $s_i = s_{\sigma_i}(y_i)$ et $y_{i+1} = \Phi_{\sigma_i}(y_i, s_{\sigma_i}(y_i)) = P_{\sigma_i}(y_i)$, on arrive à

$$dP_{\sigma_i}(y_i) = e^{A_{\sigma_i} s_i} + (A_{\sigma_i} y_{i+1} + b_{\sigma_i}) \frac{-c_{e_i}^T e^{A_{\sigma_i} s_i}}{c_{e_i}^T (A_{\sigma_i} y_{i+1} + b_{\sigma_i})}.$$

■

5.2.3 Application de Poincaré hybride

On peut alors définir l'application de Poincaré hybride. Soit $(0, \sigma_0, y_0)$ une condition initiale du système hybride avec $y_0 \in G_{e_N}$ et suffisamment proche de y_0^* . La variable continue $x(t)$ sort du domaine D_{σ_0} en un point $y_1 \in G_{e_0}$ proche de y_1^* . De proche en proche, on peut montrer que les $N+1$ premiers éléments de la séquence des modes discrets sont les mêmes que ceux de l'exécution périodique. La variable continue revient donc sur G_{e_N} en un point

$$y_{N+1} = P_{\sigma_N} \circ P_{\sigma_{N-1}} \circ \dots \circ P_{\sigma_0}(y_0).$$

L'application $P = P_{\sigma_N} \circ P_{\sigma_{N-1}} \circ \dots \circ P_{\sigma_0}$ est appelée application de Poincaré hybride. Notons qu'une exécution périodique du système hybride \mathcal{H} correspond à un point fixe de l'application de Poincaré :

$$P(y_0^*) = y_0^*.$$

Théorème 5.2.2 *L'application de Poincaré hybride P est C^∞ sur un voisinage de y_0^* . Sa différentielle est*

$$dP(y_0) = \prod_{i=N}^{i=0} dP_{\sigma_i}(y_i),$$

où

$$y_i = P_{\sigma_{i-1}} \circ \dots \circ P_{\sigma_0}(y_0).$$

Preuve : P est la composition de fonctions C^∞ , P est donc C^∞ . L'expression de dP est obtenue par l'application du théorème de composition d'applications différentiables.

■

5.3 Stabilité et calcul des exécutions périodiques

L'application de Poincaré hybride s'avère très utile pour l'analyse de stabilité et le calcul des exécutions périodiques.

5.3.1 Stabilité du cycle hybride

Le résultat présenté dans ce paragraphe est similaire à ceux exposés dans [61, 59, 88]. On dira qu'un cycle hybride est stable si pour tout y_0 dans un voisinage de y_0^* , l'exécution (τ, q, x) du système hybride ayant $(0, \sigma_0, y_0)$ pour condition initiale vérifie

$$\lim_{t \rightarrow \infty} d(x(t), \Gamma) = 0.$$

Il est clair que cela est possible si et seulement si la suite (y_0^k) des retours successifs de $x(t)$ sur la garde G_{e_N} est bien définie et converge vers y_0^* . Cette suite est définie par la relation de récurrence :

$$y_0^{k+1} = P(y_0^k), \quad y_0^0 = y_0.$$

Théorème 5.3.1 (Stabilité du cycle hybride) *Soient λ_j les valeurs propres de la matrice $dP(y_0^*)$.*

- Si pour tout j , $|\lambda_j| < 1$, alors Γ est stable. On dit que Γ est un cycle limite hybride.
- S'il existe j' , $|\lambda_{j'}| > 1$, alors Γ n'est pas stable.
- S'il existe j' , $|\lambda_{j'}| = 1$, et pour tout $j \neq j'$, $|\lambda_j| \leq 1$, alors notre méthode ne permet pas de conclure.

Preuve : y_0^* est un point fixe de l'application P . Le théorème est alors une conséquence immédiate du théorème de stabilité des points fixes par linéarisation (voir par exemple [98]).

■

Nous illustrons ce résultat en analysant la stabilité de l'oscillateur électronique à valve.

Exemple 5.3.1 (Oscillateur électronique à valve [1, 79]) *On a vu précédemment que pour $x_1(0) = i_0^*$, $x_2(0) = 0$ avec*

$$i_0^* = \frac{1}{1 - e^{-\frac{\pi h}{\omega}}},$$

le système hybride acceptait une exécution périodique.

On définit alors au voisinage du point $(i_0^, 0)$ l'application de Poincaré hybride $P(x_1, x_2)$ sur la garde $G_{(2,1)}$:*

$$P(x_1, x_2) = (P_1(x_1, x_2), P_2(x_1, x_2)).$$

Comme $P(x_1, x_2) \in G_{(2,1)}$, on a pour tout (x_1, x_2) , $P_2(x_1, x_2) = 0$.

D'autre part, on a vu précédemment que :

$$P_1(x_1, 0) = 1 + e^{-\frac{\pi h}{\omega}} + e^{-\frac{2\pi h}{\omega}} x_1.$$

Donc,

$$\frac{\partial P_1}{\partial x_1}(x_1, 0) = e^{-\frac{2\pi h}{\omega}}.$$

La différentielle de l'application de Poincaré hybride en $(x_1, 0)$ est donc de la forme

$$dP(x_1, 0) = \begin{pmatrix} e^{-\frac{2\pi h}{\omega}} & \frac{\partial P_1}{\partial x_2}(x_1, 0) \\ 0 & 0 \end{pmatrix}.$$

Les valeurs propres de $dP(i_0^*, 0)$ sont donc 0 et $e^{-\frac{2\pi h}{\omega}} < 1$. Le cycle hybride est stable.

5.3.2 Calcul du cycle hybride

Calculer le cycle hybride revient donc à calculer le point fixe y_0^* de l'application de Poincaré. Autrement dit, y_0^* est une racine de l'équation

$$Q(y) = y - P(y) = 0.$$

Si $dQ(y_0^*)$ est inversible, alors la méthode de Newton appliquée à la fonction Q converge localement vers y_0^* . On fera donc l'hypothèse que le cycle hybride n'est pas singulier : $dP(y_0^*)$ ne possède pas de valeur propre égale à 1.

Soit y_0^0 une approximation de y_0^* sur la garde G_{e_N} , la méthode de Newton appliquée à Q devient

$$\begin{aligned} y_0^{k+1} &= y_0^k - [dQ(y_0^k)]^{-1}Q(y_0^k) \\ &= y_0^k - [I - dP(y_0^k)]^{-1}(y_0^k - P(y_0^k)). \end{aligned}$$

Par conséquent, le schéma itératif est

$$y_0^{k+1} = [I - dP(y_0^k)]^{-1}(P(y_0^k) - dP(y_0^k)y_0^k) \quad (5.3.6)$$

A chaque itération de la méthode de Newton, on doit calculer les valeurs de $P(y_0^k)$ et de $dP(y_0^k)$. Le calcul de $P(y_0^k)$ est fait par simulation du système hybride. Par l'algorithme présenté dans le chapitre précédent, on simule le système hybride \mathcal{H} avec la condition initiale $(0, \sigma_0, y_0^k)$ jusqu'à ce que la variable continue $x(t)$ revienne sur la garde G_{e_N} .

Durant la simulation, les valeurs des instants et des points auxquels surviennent les transitions sont calculées. En reportant ces valeurs dans l'expression de la différentielle de l'application de Poincaré hybride donnée par les théorèmes 5.2.1 et 5.2.2, on peut alors calculer $dP(y_0^k)$.

Le calcul de y_0^{k+1} est alors possible. Ce schéma numérique est convergent, en effet le résultat classique de convergence de la méthode de Newton donne le théorème suivant.

Théorème 5.3.2 *Soit y_0^0 un point de la garde G_{e_N} , suffisamment proche de y_0^* . Soit (y_0^k) la suite définie par le schéma (5.3.6). On a alors,*

$$\lim_{k \rightarrow +\infty} y_0^k = y_0^*$$

de plus,

$$\|y_0^{k+1} - y_0^*\| = O(\|y_0^k - y_0^*\|^2).$$

Preuve : $dP(y_0^*)$ n'a pas de valeur propre égale à 1. De plus l'application de Poincaré hybride est C^∞ et sa différentielle est donc continue sur un voisinage de y_0^* . Pour y_0^k

suffisamment proche de y_0^* , $dP(y_0^k)$ n'a donc pas de valeur propre égale à 1 et la matrice $I - dP(y_0^k)$ est inversible. L'équation (5.3.6) est équivalente à

$$y_0^{k+1} = P(y_0^k) + dP(y_0^k)(y_0^{k+1} - y_0^k).$$

P est C^∞ et y_0^* est un point fixe de P , on a donc

$$\begin{aligned} y_0^* &= P(y_0^*) = P(y_0^k + (y_0^* - y_0^k)) \\ &= P(y_0^k) + dP(y_0^k)(y_0^* - y_0^k) + O(\|y_0^k - y_0^*\|^2). \end{aligned}$$

On a, par conséquent,

$$y_0^* - y_0^{k+1} = dP(y_0^k)(y_0^* - y_0^{k+1}) + O(\|y_0^k - y_0^*\|^2).$$

D'où

$$[I - dP(y_0^k)](y_0^* - y_0^{k+1}) = O(\|y_0^k - y_0^*\|^2).$$

Sur un voisinage de y_0^* , la matrice $I - dP(y)$ est inversible donc

$$\|y_0^{k+1} - y_0^*\| = O(\|y_0^k - y_0^*\|^2).$$

■

On voit, cependant, que la convergence de notre méthode est seulement locale. Ainsi, nous devons dans un premier temps avoir recours à une méthode de localisation du cycle hybride. Dans le cas général, on doit souvent utiliser un ensemble de simulations pour pouvoir deviner l'existence d'un cycle et procéder à sa localisation.

5.4 Exemple

Le système de deux réservoirs (voir figure 5.4) a été présenté dans [80] comme un exemple de cycle limite se produisant dans les systèmes hybrides.

L'étude de la stabilité du système a été faite par Rubensson et Lennartson dans [81] grâce à des fonctions de Lyapunov et par Hiskens dans [61] grâce à l'analyse de sensibilité de trajectoire. Cette dernière utilise des outils assez similaires à la différentielle de l'application de Poincaré hybride définie dans ce chapitre.

Le système est composé de deux réservoirs et deux valves. Le premier réservoir est alimenté avec un débit constant égal à Q_0 . La première valve permet d'ajouter de l'eau dans le premier réservoir. L'apport d'eau supplémentaire est alors Q_1 .

L'eau du premier réservoir passe dans le second réservoir. Le débit est proportionnel au volume d'eau v_1 dans le premier réservoir. La quantité d'eau dans le premier réservoir suit donc la loi

$$v_1'(t) = \begin{cases} -v_1(t) + Q_0, & \text{si la première valve est fermée} \\ -v_1(t) + Q_0 + Q_1, & \text{si la première valve est ouverte.} \end{cases}$$

La seconde valve permet de vider le second réservoir. Le débit est proportionnel au volume d'eau v_2 dans le deuxième réservoir. On prélève également de l'eau dans le deuxième

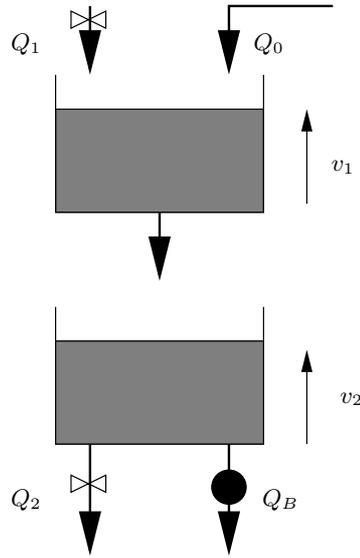


FIG. 5.4: Système de deux réservoirs

réservoir à un débit constant Q_B . La quantité d'eau dans le deuxième réservoir suit donc la loi

$$v_2'(t) = \begin{cases} v_1(t) & -Q_B, \text{ si la deuxième valve est fermée} \\ v_1(t) - v_2(t) & -Q_B, \text{ si la deuxième valve est ouverte.} \end{cases}$$

L'objectif est de garder les volumes d'eau dans certaines limites autour de niveaux de références v_1^* et v_2^* en utilisant une stratégie *feedback* d'ouverture/fermeture des valves.

Nous disposons donc d'un total de quatre modes discrets pour notre système hybride linéaire par morceaux (voir tableau 5.1).

État du système hybride	valve 1	valve 2
$q(t) = 1$	fermée	fermée
$q(t) = 2$	ouverte	fermée
$q(t) = 3$	fermée	ouverte
$q(t) = 4$	ouverte	ouverte

TAB. 5.1: Les états discrets du système hybride des deux réservoirs

On introduit les variables $x_1(t) = v_1(t) - v_1^*$ et $x_2(t) = v_2(t) - v_2^*$. La variable continue du système hybride est $x(t) = (x_1(t), x_2(t))$. Le but est donc de garder les variables $x_1(t)$ et $x_2(t)$ proche de 0.

Les dynamiques continues sont données par

$$A_1 = \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix}, \quad b_1 = \begin{pmatrix} -2 \\ 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix}, \quad b_2 = \begin{pmatrix} 3 \\ 0 \end{pmatrix},$$

$$A_3 = \begin{pmatrix} -1 & 0 \\ 1 & -1 \end{pmatrix}, \quad b_3 = \begin{pmatrix} -2 \\ -5 \end{pmatrix}, \quad A_4 = \begin{pmatrix} -1 & 0 \\ 1 & -1 \end{pmatrix}, \quad b_4 = \begin{pmatrix} 3 \\ -5 \end{pmatrix}.$$

On définit la stratégie d'ouverture/fermeture des valves en définissant les contraintes du système hybride linéaire par morceaux.

Lorsque l'on est dans l'état 1, les deux valves sont fermées. Le niveau dans le premier réservoir monte alors qu'il descend dans le deuxième. Si x_1 atteint -1 , on ouvre le premier robinet pour remplir le premier réservoir. Si x_2 atteint 1 , on ouvre le deuxième robinet pour vider le deuxième réservoir. On a donc

$$C_{(1,2)} = \{x_1 \geq -1\}, \quad C_{(1,3)} = \{x_2 \leq 1\}.$$

Lorsque l'on est dans l'état 2, la première valve est ouverte et la deuxième est fermée; les deux réservoirs se remplissent. Lorsque le niveau dans le deuxième réservoir atteint 1 , on ferme la première valve et on ouvre la seconde :

$$C_{(2,3)} = \{x_2 \leq 1\}.$$

Lorsque l'on est dans l'état 3, la première valve est fermée et la deuxième ouverte; les deux réservoirs se vident. Lorsque le niveau dans le premier réservoir atteint -1 , on ouvre la première valve pour remplir le premier réservoir. Lorsque le niveau dans le deuxième réservoir atteint 0 , on ferme la deuxième valve.

$$C_{(3,4)} = \{x_1 \geq -1\}, \quad C_{(3,1)} = \{x_2 \geq 0\}.$$

Enfin, dans l'état 4, les deux valves sont ouvertes. Le premier réservoir se remplit, le deuxième se vide. Lorsque le niveau dans le premier réservoir atteint 1 on ferme la première valve. Lorsque le niveau dans le second reservoir atteint 0 on ferme la deuxième valve.

$$C_{(4,3)} = \{x_1 \leq 1\}, \quad C_{(4,2)} = \{x_2 \geq 0\}.$$

Sur la figure 5.5, nous avons tracé un exemple d'exécution du système de deux réservoirs. Les droites tracées en pointillés représentent les gardes du système. Il semble que la variable $x(t)$ du système tende vers un cycle limite hybride. La séquence d'états discrets correspondant au cycle hybride est $\{1, 2, 3\}$.

Pour initialiser l'algorithme, nous devons choisir un point proche du cycle sur la garde $G_{(3,1)}$. $y_0^0 = (1, 0)^T$ semble être un point initial raisonnable. Les résultats du calcul du cycle hybride effectué par notre algorithme sont montrés dans le tableau 5.2.

On voit que la convergence de notre méthode est bien quadratique. Seulement trois itérations sont requises pour obtenir une approximation du cycle limite hybride avec une précision proche de celle autorisée par l'arithmétique flottante utilisée par matlab.

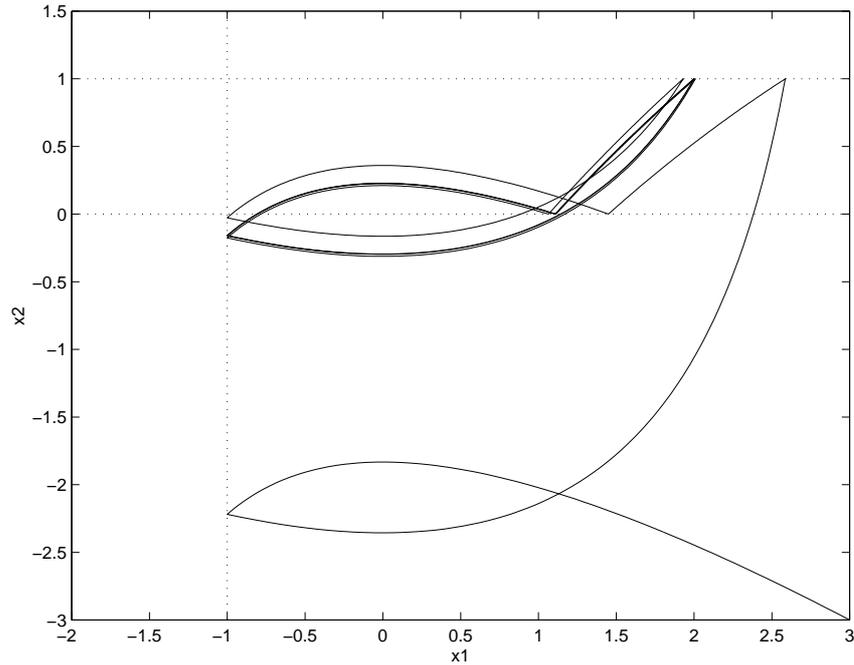


FIG. 5.5: Exemple d'exécution du système hybride

k	$(y_0^k)^T$	$\ P(y_0^k) - y_0^k\ $
0	[1.0000000000000000 0]	0.1195483220052789
1	[1.1086673728553014 0]	0.0005631232587389
2	[1.1081603057366756 0]	0.0000000125312090
3	[1.1081602944523914 0]	0.0000000000000390

TAB. 5.2: Résultats du calcul du cycle hybride

A la dernière itération, la différentielle de l'application de Poincaré hybride a pour matrice

$$dP(y_0^k) = \begin{pmatrix} -0.1105 & -0.3099 \\ 0 & 0 \end{pmatrix}$$

dont les valeurs propres sont 0 et -0.1105. Par conséquent, le cycle hybride est stable. De plus sa période est d'environ 2.7732 et les temps passés dans chaque état sont 1.1340 pour l'état 1, 1.3868 pour l'état 2 et 0.2524 pour l'état 3. Sur la figure 5.6, le cycle limite hybride du système de deux réservoirs est représenté.

Remarque 5.4.1 *Il est possible, pour un système plan, d'avoir une des valeurs propres de la différentielle qui est négative. Ceci ne se produit jamais dans les systèmes purement continus et est rendu possible ici car les trajectoires parcourues dans \mathbb{R}^2 par la variable continue du système peuvent se croiser.*

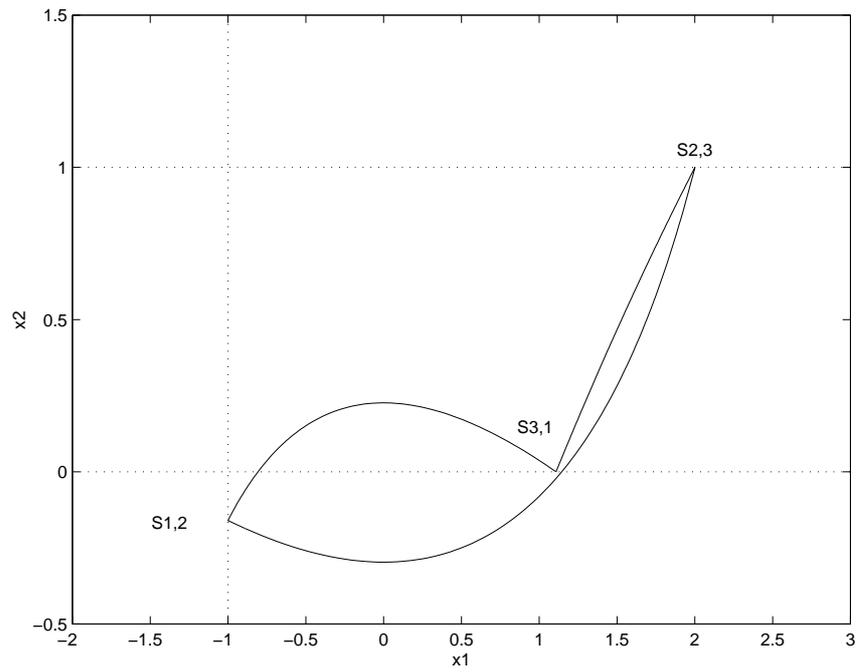


FIG. 5.6: Cycle hybride du système de deux réservoirs

Troisième partie

Analyse d'Atteignabilité

Chapitre 6

Atteignabilité des Systèmes Dynamiques Linéaires

Dans la partie précédente, nous avons vu comment réaliser l'analyse de systèmes hybrides linéaires par morceaux par le calcul de ses exécutions. Cependant, il est fréquent que la dynamique continue d'un système hybride ne soit connue que de manière approximative :

- soit parce que les valeurs des paramètres du système sont mesurées avec une marge d'incertitude;
- soit parce que dans le processus de modélisation, on néglige l'effet de certains phénomènes n'ayant qu'une action limitée sur le comportement du système.

Cette incertitude sur la dynamique continue du système peut être intégrée au modèle en ajoutant, aux champs de vecteurs linéaires, un terme de perturbation. Dans chaque mode $q \in \mathcal{Q}$, l'évolution de la variable continue est alors donnée par une inclusion différentielle (voir par exemple les travaux de Aubin [8, 9]) :

$$x'(t) = A_q x(t) + b_q + u(t), \|u(t)\| \leq \mu_q \quad (6.0.1)$$

où $\|\cdot\|$ dénote une norme polyédrique¹. Il est facile de voir que le système hybride résultant, \mathcal{H} , est linéaire par morceaux.

Cette modélisation a l'avantage d'être plus réaliste; en revanche, pour une condition initiale donnée, le système hybride accepte une infinité d'exécutions maximales. Pour étudier l'ensemble des comportements possibles du système, il faut alors définir la notion d'ensemble atteignable.

Soit $I \subseteq \mathcal{Q} \times R^n$, un ensemble de conditions initiales possibles, l'ensemble atteignable $\mathcal{R}(I)$ du système hybride \mathcal{H} pour l'ensemble de conditions initiales I est l'ensemble des couples $(\sigma, y) \in \mathcal{Q} \times R^n$, tels qu'il existe une exécution (τ, q, x) acceptée par \mathcal{H} vérifiant

$$(q(t_0), x(t_0)) \in I \text{ et } \exists t \in \tau, q(t) = \sigma, x(t) = y.$$

Typiquement, le calcul de cet ensemble est requis pour la vérification des systèmes hybrides. Le travail de vérification des systèmes hybrides est une tâche difficile. Il s'agit de garantir algorithmiquement qu'un système hybride vérifie une propriété donnée.

¹Une norme est dite polyédrique si sa boule unité est un polytope. Par exemple, la norme 1 et la norme ∞ sont polyédriques.

On peut citer, par exemple, le problème suivant (vérification d'une propriété de sûreté). Etant donné un ensemble de conditions initiales $I \subseteq \mathcal{Q} \times R^n$ et un ensemble de panne $U \subseteq \mathcal{Q} \times R^n$, il faut prouver que les exécutions acceptées par \mathcal{H} pour les conditions initiales I n'atteignent jamais l'ensemble critique U :

$$\mathcal{R}(I) \cap U = \emptyset.$$

L'importance du problème a motivé de nombreux travaux sur le sujet. Pour une classe particulière de systèmes hybrides constants par morceaux, appelés automates temporisés, des méthodes et des outils pour le calcul exact de l'ensemble atteignable ont été développés [58, 99]. Pour des systèmes avec des dynamiques continues plus complexes, le calcul exact de l'ensemble atteignable est difficile. Même pour des systèmes hybrides linéaires par morceaux, ceci est possible uniquement pour certaines matrices en fonction de leurs valeurs propres [67].

D'un autre côté, plusieurs méthodes d'approximation de l'ensemble atteignable ont été développées, [2, 11, 21, 23, 54, 75]. Ces méthodes approchent l'ensemble atteignable en utilisant différentes représentations classiques des ensembles : union de polytopes, d'hyperrectangles orthogonaux, ellipsoïdes, ensembles algébriques.

Dans cette partie, nous considérerons la classe des systèmes hybrides linéaires par morceaux où les dynamiques continues sont données par des inclusions différentielles telles que (6.0.1). Un prérequis nécessaire à l'analyse d'atteignabilité de tels systèmes hybrides est le calcul de l'ensemble atteignable d'un système dynamique linéaire soumis à des perturbations.

Dans ce chapitre, nous commencerons par étudier les deux approches existantes pour le calcul d'une approximation de l'ensemble atteignable d'un système dynamique linéaire autonome. Nous verrons, ensuite, comment une de ces deux méthodes peut être adaptée au cas de systèmes soumis à des perturbations. Mais d'abord, il nous faut préciser la notion d'approximation pour les ensembles.

6.1 Approximation d'ensembles

Il est souvent intéressant d'approcher un ensemble E par un autre ensemble F tel que tout élément de E est aussi dans F .

Définition 6.1.1 (Approximation Conservative) *On dit que F est une approximation conservative de l'ensemble E si $E \subseteq F$.*

Pour évaluer la qualité de cette approximation, il est nécessaire d'introduire une notion de distance sur les ensembles.

Définition 6.1.2 (Distance de Hausdorff [39]) *Soit $\|\cdot\|$ une norme sur \mathbb{R}^n . L'application $d_H : 2^{\mathbb{R}^n} \times 2^{\mathbb{R}^n} \rightarrow \mathbb{R}^+$ définie par*

$$d_H(E, F) = \max \left(\sup_{x \in E} \inf_{y \in F} \|x - y\|, \sup_{y \in F} \inf_{x \in E} \|x - y\| \right)$$

est une distance sur l'ensemble des parties de \mathbb{R}^n .

La distance de Hausdorff vérifie certaines propriétés intéressantes :

Proposition 6.1.1 (Propriétés de la distance de Hausdorff [39])

1. Si F est une approximation conservative de E , alors

$$d_H(E, F) = \sup_{y \in F} \inf_{x \in E} \|x - y\|.$$

2. Soient E_1, E_2, F_1, F_2 des parties de \mathbb{R}^n :

$$d_H(E_1, F_1) \leq \delta \text{ et } d_H(E_2, F_2) \leq \delta \implies d_H(E_1 \cup E_2, F_1 \cup F_2) \leq \delta.$$

6.2 Atteignabilité des systèmes linéaires autonomes

On considère le système dynamique linéaire autonome suivant :

$$x'(t) = Ax(t) + b, \quad x(t) \in \mathbb{R}^n.$$

Notons Φ le flot associé à cette équation différentielle.

$$\Phi(y, t) = e^{At}y + \int_0^t e^{A(t-s)}b \, ds.$$

On se donne un ensemble de conditions initiales $I \subseteq \mathbb{R}^n$. Notons

$$\Phi(I, t) = \{\Phi(y, t), y \in I\}.$$

On cherche à calculer $\mathcal{R}(I)$, l'ensemble atteignable par le système dynamique pour l'ensemble de conditions initiales I . Cet ensemble est défini par :

$$\mathcal{R}(I) = \{\Phi(y, s), y \in I, s \in \mathbb{R}^+\} = \bigcup_{s \in \mathbb{R}^+} \Phi(I, s).$$

Le problème de cette caractérisation, est qu'en général, il est impossible de vérifier si un point de \mathbb{R}^n appartient à cet ensemble. On cherche donc à approcher cet ensemble de manière conservative : $\mathcal{R}(I)$ est inclus dans l'approximation. Plus généralement, on cherchera à calculer une approximation de l'ensemble atteignable par le système dynamique sur l'intervalle de temps $[0, t]$.

Définition 6.2.1 L'ensemble atteignable du système dynamique sur l'intervalle $[0, t]$, pour un ensemble de conditions initiales I est noté $\mathcal{R}_{[0,t]}(I)$ et est défini par la relation

$$\mathcal{R}_{[0,t]}(I) = \bigcup_{s \in [0,t]} \Phi(I, s).$$

Pour la classe des systèmes linéaires, des méthodes [2, 21] et des outils (d/dt [4, 25] ou CheckMate [22, 86]) efficaces, ont été proposés pour le calcul des ensembles atteignables. Chacune de ces approches utilise une des propriétés de décomposition de l'ensemble atteignable données par la proposition suivante.

Proposition 6.2.1 Soient $r > 0$, $N \in \mathbb{N}$, on note $t = Nr$,

$$\mathcal{R}_{[0,t]}(I) = \bigcup_{k=0}^{k=N-1} \mathcal{R}_{[0,r]}(\Phi(I, kr)), \quad (6.2.2)$$

$$\mathcal{R}_{[0,t]}(I) = \bigcup_{k=0}^{k=N-1} \Phi(\mathcal{R}_{[0,r]}(I), kr). \quad (6.2.3)$$

Preuve : On a clairement

$$\mathcal{R}_{[0,t]}(I) = \bigcup_{k=0}^{k=N-1} \mathcal{R}_{[kr, (k+1)r]}(I).$$

Remarquons que l'ensemble atteignable par le système dynamique sur l'intervalle $[kr, (k+1)r]$, $\mathcal{R}_{[kr, (k+1)r]}(I)$, peut être calculé de deux manières différentes. Soit x un élément de cet ensemble :

$$\begin{aligned} \exists y \in I, s \in [kr, (k+1)r], x &= \Phi(y, s) \\ &= \Phi(\Phi(y, kr), s - kr) \\ &= \Phi(\Phi(y, s - kr), kr). \end{aligned}$$

On peut donc :

- soit calculer l'ensemble atteignable sur l'intervalle $[0, r]$ pour l'ensemble de conditions initiales $\Phi(I, kr)$:

$$\mathcal{R}_{[kr, (k+1)r]}(I) = \mathcal{R}_{[0,r]}(\Phi(I, kr));$$

- soit calculer l'image par le flot à l'instant kr de l'ensemble atteignable sur l'intervalle $[0, r]$:

$$\mathcal{R}_{[kr, (k+1)r]}(I) = \Phi(\mathcal{R}_{[0,r]}(I), kr).$$

■

On voit donc, que deux approches distinctes sont possibles pour le calcul de l'ensemble atteignable sur l'intervalle $[0, t]$.

Nous souhaitons calculer une approximation conservative de l'ensemble $\mathcal{R}_{[0,t]}(I)$ sous la forme d'une union de polytopes. La proposition 6.2.1 montre qu'une telle approximation peut être calculée à partir de deux fonctions :

- Une fonction qui à un polytope I associe son image par le flot $\Phi(I, r)$;
- Une fonction qui à un polytope I associe un polytope $\hat{\mathcal{R}}_{[0,r]}(I)$, approximation conservative de l'ensemble atteignable $\mathcal{R}_{[0,r]}(I)$, pour r arbitrairement petit.

6.2.1 Calcul de l'image par le flot

L'image d'un polytope I par le flot est un polytope et est aisément calculable.

Proposition 6.2.2 Soit I un polytope, I est l'enveloppe convexe d'un ensemble fini de points $\{y_1, \dots, y_p\}$, l'ensemble $\Phi(I, r)$ est l'enveloppe convexe de l'ensemble de points $\{\Phi(y_1, r), \dots, \Phi(y_p, r)\}$.

Preuve : Soit y un point de I , il existe p réels positifs $\{\lambda_1, \dots, \lambda_p\}$ tels que,

$$y = \sum_{i=1}^{i=p} \lambda_i y_i, \quad \sum_{i=1}^{i=p} \lambda_i = 1.$$

Comme l'application qui à y associe $\Phi(y, r)$ est une application affine, on a

$$\Phi(y, r) = \sum_{i=1}^{i=p} \lambda_i \Phi(y_i, r).$$

■

Le calcul de l'image d'un polytope par le flot est donc simple et peut être implémenté efficacement.

6.2.2 Approximation conservative de $\mathcal{R}_{[0,r]}(I)$

Le calcul d'un polytope, $\hat{\mathcal{R}}_{[0,r]}(I)$, contenant l'ensemble atteignable $\mathcal{R}_{[0,r]}(I)$ est beaucoup plus délicat que le calcul de l'image par le flot. La méthode présentée ici est similaire à celle utilisée par d/dt, on pourra aussi consulter la thèse de T. Dang [24].

Soit I un polytope, observons d'abord que I et $\Phi(I, r)$ sont inclus dans $\mathcal{R}_{[0,r]}(I)$. Donc,

$$\mathcal{R}_{[0,r]}(I) \subseteq \hat{\mathcal{R}}_{[0,r]}(I) \implies I \subseteq \hat{\mathcal{R}}_{[0,r]}(I) \text{ et } \Phi(I, r) \subseteq \hat{\mathcal{R}}_{[0,r]}(I).$$

De plus, I est l'enveloppe convexe d'un ensemble fini de points $\{y_1, \dots, y_p\}$. D'après la proposition 6.2.2, $\Phi(I, r)$ est l'enveloppe convexe de l'ensemble $\{\Phi(y_1, r), \dots, \Phi(y_p, r)\}$. Soit J l'enveloppe convexe de l'ensemble de points $\{y_1, \dots, y_p, \Phi(y_1, r), \dots, \Phi(y_p, r)\}$. Puisque $\hat{\mathcal{R}}_{[0,r]}(I)$ doit être un polytope contenant cet ensemble de points, on a nécessairement,

$$J \subseteq \hat{\mathcal{R}}_{[0,r]}(I).$$

On définit également l'ensemble J' :

$$J' = \left\{ x + \frac{s}{r}(\Phi(x, r) - x), x \in I, s \in [0, r] \right\}.$$

J' est l'ensemble des segments reliant les points de I à leur image par le flot à l'instant r . On a donc $J' \subseteq J$. De plus, intuitivement, pour r petit, J' est une bonne approximation de $\mathcal{R}_{[0,r]}(I)$. A l'aide de la distance de Hausdorff, on peut quantifier la qualité de cette approximation.

Lemme 6.2.1 $d_H(\mathcal{R}_{[0,r]}(I), J') \leq \varepsilon_r$ avec

$$\varepsilon_r = \frac{r^2}{2} e^{r\|A\|} \|A\| \sup_{x \in I} \|Ax + b\|.$$

Preuve : Soit y un élément de l'ensemble atteignable $\mathcal{R}_{[0,r]}(I)$:

$$\exists x \in I, s \in [0, r], y = \Phi(x, s).$$

Remarquons que, par la formule de Taylor,

$$y = x + \sum_{k=1}^{k=\infty} \frac{s^k}{k!} A^{k-1} (Ax + b).$$

Soit z le point de J' , défini par $z = x + \frac{s}{r} (\Phi(x, r) - x)$. On a de même

$$z = x + \sum_{k=1}^{k=\infty} \frac{sr^{k-1}}{k!} A^{k-1} (Ax + b).$$

D'où,

$$z - y = \sum_{k=2}^{k=\infty} \frac{s(r^{k-1} - s^{k-1})}{k!} A^{k-1} (Ax + b).$$

Pour s dans l'intervalle $[0, r]$, $s(r^{k-1} - s^{k-1}) \leq r^k$, donc

$$\begin{aligned} \|z - y\| &\leq \sum_{k=2}^{k=\infty} \frac{r^k}{k!} \|A\|^{k-1} \|Ax + b\| \\ &\leq \frac{1}{\|A\|} \left(e^{r\|A\|} - 1 - r\|A\| \right) \|Ax + b\| \\ &\leq \frac{r^2}{2} e^{r\|A\|} \|A\| \|Ax + b\| \leq \varepsilon_r. \end{aligned}$$

De la même manière, on peut montrer que, pour tout $z \in J'$, il existe $y \in \mathcal{R}_{[0,r]}(I)$, tel que $\|z - y\| \leq \varepsilon_r$. ■

On note $\mathcal{B}(0, \varepsilon_r)$ la boule centrée en 0 et de rayon ε_r , le symbole \oplus dénote la somme de Minkowski [101]. D'après le lemme précédent, il est clair que

$$\mathcal{R}_{[0,r]}(I) \subseteq J' \oplus \mathcal{B}(0, \varepsilon_r).$$

On a donc une approximation conservative de l'ensemble $\mathcal{R}_{[0,r]}(I)$. Malheureusement, l'ensemble J' est en général non-convexe (voir figure 6.1) et donc l'ensemble $J' \oplus \mathcal{B}(0, \varepsilon_r)$ ne peut convenir.

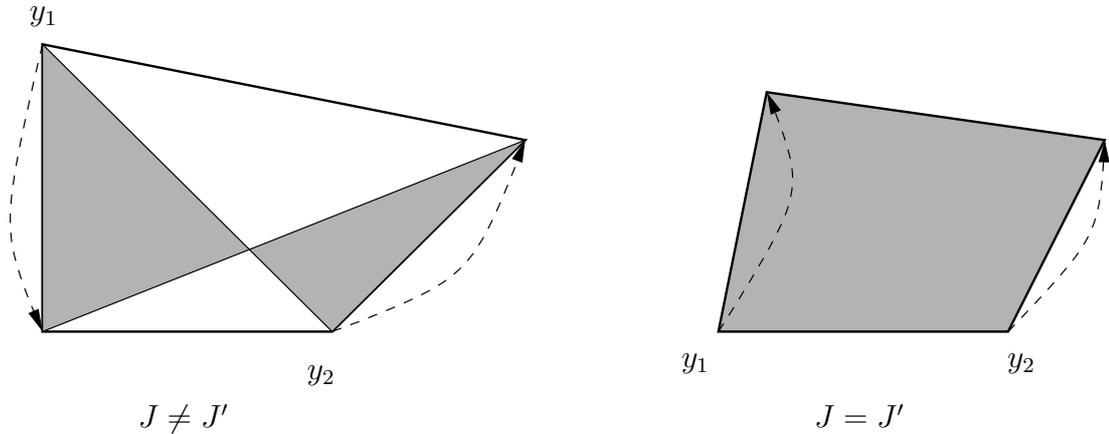


FIG. 6.1: L'ensemble J' est une bonne approximation de $\mathcal{R}_{[0,r]}(I)$ mais n'est pas toujours un polytope. Le plus petit polytope contenant J' est J .

Le polytope J est l'enveloppe convexe de J' , et en pratique, il est fréquent que l'on ait $J = J'$. Par conséquent, il semble que le choix de

$$\hat{\mathcal{R}}_{[0,r]}(I) = J \oplus \mathcal{B}(0, \varepsilon_r)$$

soit raisonnable.

Le polytope $\hat{\mathcal{R}}_{[0,r]}(I)$ peut donc être calculé par l'algorithme suivant.

Données : Un polytope I

I est l'enveloppe convexe de $\{y_1, \dots, y_p\}$.

Résultat : Une approximation conservatrice $\hat{\mathcal{R}}_{[0,r]}(I)$ de l'ensemble $\mathcal{R}_{[0,r]}(I)$.

Calculer J l'enveloppe convexe de $\{y_1, \dots, y_p, \Phi(y_1, r), \dots, \Phi(y_p, r)\}$;

$\hat{\mathcal{R}}_{[0,r]}(I) := J \oplus \mathcal{B}(0, \varepsilon_r)$;

Algorithme 3: Calcul de $\hat{\mathcal{R}}_{[0,r]}(I)$

Proposition 6.2.3 Soit $\hat{\mathcal{R}}_{[0,r]}(I)$ l'ensemble calculé par l'algorithme 3.

Si $\|\cdot\|$ est une norme polyédrique, alors $\hat{\mathcal{R}}_{[0,r]}(I)$ est un polytope et

$$\mathcal{R}_{[0,r]}(I) \subseteq \hat{\mathcal{R}}_{[0,r]}(I).$$

De plus (voir figure 6.1) :

$$- \text{Si } J = J', \delta_r(I) = d_H(\mathcal{R}_{[0,r]}(I), \hat{\mathcal{R}}_{[0,r]}(I)) \leq 2\varepsilon_r$$

$$- \text{Si } J \neq J', \delta_r(I) = d_H(\mathcal{R}_{[0,r]}(I), \hat{\mathcal{R}}_{[0,r]}(I)) \leq 2\varepsilon_r + re^{r\|A\|} \sup_{x \in I} \|Ax + b\|$$

avec

$$\varepsilon_r = \frac{r^2}{2} e^{r\|A\|} \|A\| \sup_{x \in I} \|Ax + b\|.$$

Preuve : J est un polytope, si $\|\cdot\|$ est une norme polyédrique, $\mathcal{B}(0, \varepsilon_r)$ est un polytope et $J \oplus \mathcal{B}(0, \varepsilon_r)$ est donc également un polytope [101].

De plus, $J' \subseteq J$, et d'après le lemme 6.2.1, on a

$$\mathcal{R}_{[0,r]}(I) \subseteq J' \oplus \mathcal{B}(0, \varepsilon_r) \subseteq J \oplus \mathcal{B}(0, \varepsilon_r) = \hat{\mathcal{R}}_{[0,r]}(I).$$

Enfin,

$$\begin{aligned} d_H(\mathcal{R}_{[0,r]}(I), \hat{\mathcal{R}}_{[0,r]}(I)) &\leq d_H(\mathcal{R}_{[0,r]}(I), J' \oplus \mathcal{B}(0, \varepsilon_r)) + d_H(J' \oplus \mathcal{B}(0, \varepsilon_r), \hat{\mathcal{R}}_{[0,r]}(I)) \\ &\leq d_H(\mathcal{R}_{[0,r]}(I), J') + d_H(J', J' \oplus \mathcal{B}(0, \varepsilon_r)) + \\ &\quad d_H(J' \oplus \mathcal{B}(0, \varepsilon_r), J \oplus \mathcal{B}(0, \varepsilon_r)) \\ &\leq 2\varepsilon_r + d_H(J', J). \end{aligned}$$

Supposons $J \neq J'$, soit $y \in J$, il existe $2p$ réels positifs $\{\alpha_1, \dots, \alpha_p, \beta_1, \dots, \beta_p\}$ tels que

$$y = \sum_{i=1}^{i=p} \alpha_i y_i + \sum_{i=1}^{i=p} \beta_i \Phi(y_i, r), \quad \sum_{i=1}^{i=p} \alpha_i + \sum_{i=1}^{i=p} \beta_i = 1.$$

Soit z l'élément de $I \subseteq J'$ défini par

$$z = \sum_{i=1}^{i=p} (\alpha_i + \beta_i) y_i,$$

On a,

$$y - z = \sum_{i=1}^{i=p} \beta_i (\Phi(y_i, r) - y_i) = \sum_{i=1}^{i=p} \beta_i \sum_{k=1}^{k=\infty} \frac{r^k}{k!} A^{k-1} (Ay_i + b).$$

D'où

$$\begin{aligned} \|y - z\| &\leq \sup_{x \in I} \|Ax + b\| \sum_{k=1}^{k=\infty} \frac{r^k}{k!} \|A\|^{k-1} \leq \sup_{x \in I} \|Ax + b\| \frac{1}{\|A\|} (e^{r\|A\|} - 1) \\ &\leq r e^{r\|A\|} \sup_{x \in I} \|Ax + b\|. \end{aligned}$$

Donc, $d_H(J', J) \leq r e^{r\|A\|} \sup_{x \in I} \|Ax + b\|$. ■

Ainsi, si l'on se trouve dans le cas où $J \neq J'$, la fonction d'approximation $\hat{\mathcal{R}}_{[0,r]}$ est en $O(r)$. Si l'on est dans le cas, relativement fréquent, où $J = J'$, alors l'approximation est en $O(r^2)$, ce qui est nettement mieux. Dans [24], on peut voir comment, en subdivisant l'ensemble I , on peut se ramener au cas favorable. Cette transformation engendre bien entendu un coût supplémentaire non négligeable.

Présentons maintenant les deux approches pour le calcul d'une approximation de l'ensemble atteignable $\mathcal{R}_{[0,t]}(I)$.

6.2.3 L'approche de d/dt

Développé au laboratoire Verimag de Grenoble par Dang, Maler et Asarin, d/dt [4, 25] est un outil de vérification des systèmes continus et hybrides basé sur le calcul des ensembles atteignables.

L'algorithme 4, utilisé par d/dt, repose sur la décomposition de l'ensemble atteignable donnée par l'équation (6.2.2).

Données : Un polytope I .

Résultat : Une approximation conservative R_N de l'ensemble $\mathcal{R}_{[0,t]}(I)$.

$I_0 := I; R_0 := I;$

pour $k = 0 \dots N - 1$ **faire**

$R_{k+1} := R_k \cup \hat{\mathcal{R}}_{[0,r]}(I_k);$
 $I_{k+1} := \Phi(I_k, r);$

fin

Algorithme 4: Calcul de l'ensemble atteignable (approche d/dt)

L'ensemble I_k est égal à $\Phi(I, kr)$, l'image de I par le flot à l'instant kr , R_k est une approximation conservative de l'ensemble atteignable du système dynamique $\mathcal{R}_{[0,kr]}(I)$ sur l'intervalle de temps $[0, kr]$.

Une illustration des premières étapes de l'algorithme est montrée sur la figure 6.2 :

- Tout d'abord, on calcule une approximation conservative $R_1 = \hat{\mathcal{R}}_{[0,r]}(I_0)$ de l'ensemble atteignable $\mathcal{R}_{[0,r]}(I)$.
- Puis on calcule l'image de l'ensemble de conditions initiales I par le flot à l'instant $r : I_1 = \Phi(I_0, r)$.
- Ensuite, on recommence le même procédé pour le nouvel ensemble de conditions initiales I_1 . L'ensemble $R_2 = R_1 \cup \hat{\mathcal{R}}_{[0,r]}(I_1)$ est alors une approximation conservative de l'ensemble atteignable $\mathcal{R}_{[0,2r]}(I)$.

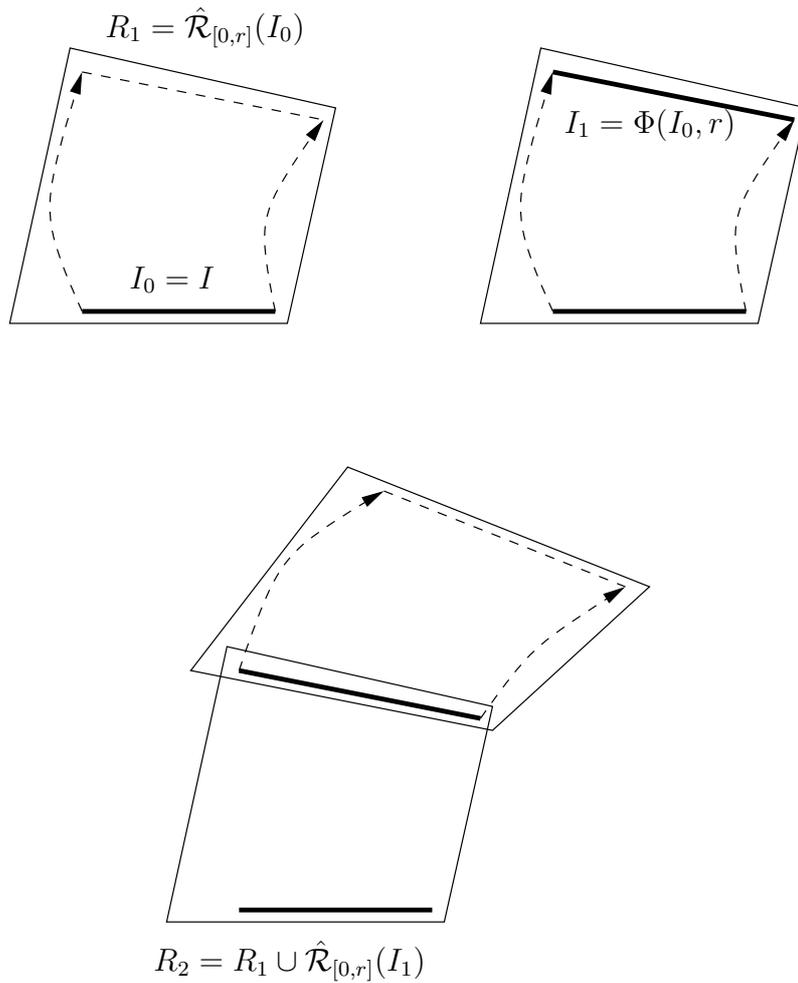


FIG. 6.2: Illustration de la méthode de calcul de l'ensemble atteignable par d/dt

Théorème 6.2.1 Soit R_N l'ensemble calculé par l'algorithme 4 :

1. (Approximation conservative) $\mathcal{R}_{[0,t]}(I) \subseteq R_N$.
2. (Convergence de l'approximation)

$$d_H(\mathcal{R}_{[0,t]}(I), R_N) \leq r(1 + r\|A\|)e^{t\|A\|} \sup_{x \in I} \|Ax + b\|.$$

Preuve : Soit y un point de $\mathcal{R}_{[0,t]}(I)$. D'après l'équation (6.2.2), il existe $k \in \{0, \dots, N-1\}$, tel que $y \in \mathcal{R}_{[0,r]}(\Phi(I, kr))$.

$$\Phi(I, kr) = I_k \text{ et } \mathcal{R}_{[0,r]}(I_k) \subseteq \hat{\mathcal{R}}_{[0,r]}(I_k) \subseteq R_N.$$

D'où $y \in R_N$.

Calculons l'erreur d'approximation produite par l'outil d/dt. On a

$$R_N = \bigcup_{k=0}^{k=N-1} \hat{\mathcal{R}}_{[0,r]}(\Phi(I, kr)).$$

D'après la proposition 6.2.3, pour tout $k \in \{0, \dots, N-1\}$,

$$d_H(\mathcal{R}_{[0,r]}(\Phi(I, kr)), \hat{\mathcal{R}}_{[0,r]}(\Phi(I, kr))) = \delta_r(\Phi(I, kr)).$$

En utilisant la deuxième propriété de la distance de Hausdorff donnée par la proposition 6.1.1,

$$d_H \left(\bigcup_{k=0}^{k=N-1} \mathcal{R}_{[0,r]}(\Phi(I, kr)), \bigcup_{k=0}^{k=N-1} \hat{\mathcal{R}}_{[0,r]}(\Phi(I, kr)) \right) \leq \max_{k=0}^{k=N-1} \delta_r(\Phi(I, kr)).$$

D'où,

$$\begin{aligned} d_H(\mathcal{R}_{[0,t]}(I), R_N) &\leq \max_{k=0}^{k=N-1} \delta_r(\Phi(I, kr)) \\ &\leq r e^{r\|A\|} (1 + r\|A\|) \sup_{x \in I, 1 \leq k \leq N-1} \|A\Phi(x, kr) + b\|. \end{aligned}$$

De plus,

$$A\Phi(x, kr) + b = \frac{\partial \Phi}{\partial t}(x, kr) = e^{krA}(Ax + b).$$

■

6.2.4 L'approche de CheckMate

La boîte à outils Checkmate [22, 86], développée à l'université de Carnegie Mellon de Pittsburgh, permet également la vérification des systèmes hybrides grâce au calcul des ensembles atteignables.

L'algorithme 5, utilisé par CheckMate, repose sur la décomposition de l'ensemble atteignable donnée par l'équation (6.2.3).

Données : Un polytope I .

Résultat : Une approximation conservative R_N de l'ensemble $\mathcal{R}_{[0,t]}(I)$.

$P_1 := \hat{\mathcal{R}}_{[0,r]}(I); R_1 := P_1;$

pour $k = 1 \dots N - 1$ **faire**

$P_{k+1} := \Phi(P_k, r);$
 $R_{k+1} := R_k \cup P_{k+1};$

fin

Algorithme 5: Calcul de l'ensemble atteignable (approche CheckMate)

P_{k+1} est une approximation conservative de $\mathcal{R}_{[kr, (k+1)r]}(I)$, l'ensemble atteignable du système dynamique sur l'intervalle $[kr, (k+1)r]$. R_{k+1} est une approximation conservative de $\mathcal{R}_{[0, (k+1)r]}(I)$, l'ensemble atteignable du système dynamique sur l'intervalle $[0, (k+1)r]$.

Une illustration des premières étapes de l'algorithme est montrée sur la figure 6.3 :

- La première étape de l'algorithme est la même que pour l'algorithme précédent; on calcule une approximation conservative $R_1 = P_1 = \hat{\mathcal{R}}_{[0,r]}(I)$ de l'ensemble atteignable $\mathcal{R}_{[0,r]}(I)$.
- Ensuite, on se contente de propager l'ensemble atteignable; $P_2 = \Phi(P_1, r)$ donne une approximation conservative de $\mathcal{R}_{[r,2r]}(I)$. $R_2 = R_1 \cup P_2$ est alors une approximation conservative de $\mathcal{R}_{[0,2r]}(I)$.

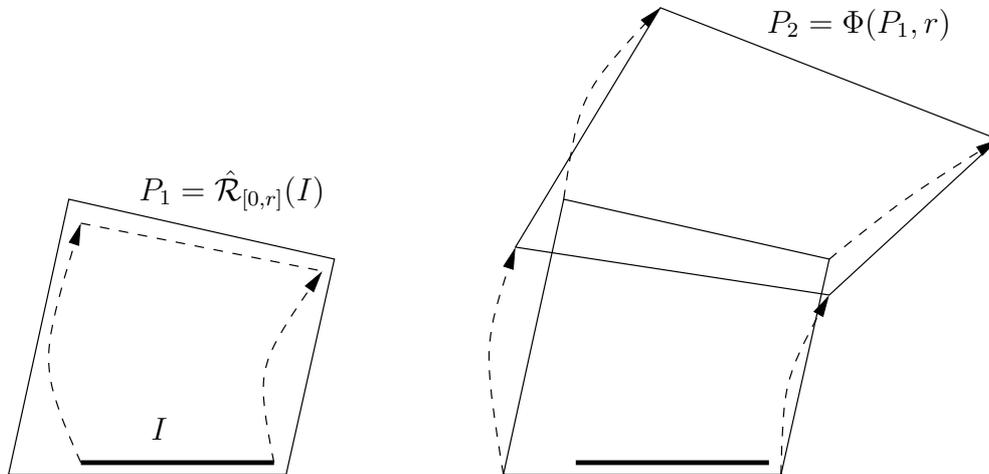


FIG. 6.3: Illustration de la méthode de calcul de l'ensemble atteignable par CheckMate

Théorème 6.2.2 Soit R_N l'ensemble calculé par l'algorithme 5 :

1. (Approximation conservative) $\mathcal{R}_{[0,t]}(I) \subseteq R_N$.
2. (Convergence de l'approximation)

$$d_H(\mathcal{R}_{[0,t]}(I), R_N) \leq e^{\|A\|(t-r)} \delta_r(I).$$

Preuve : Soit y un point de $\mathcal{R}_{[0,t]}(I)$, d'après l'équation (6.2.3), il existe $k \in \{0, \dots, N-1\}$, tel que $y \in \Phi(\mathcal{R}_{[0,r]}(I), kr)$. Or,

$$\mathcal{R}_{[0,r]}(I) \subseteq \hat{\mathcal{R}}_{[0,r]}(I) = P_1.$$

Donc,

$$\Phi(\mathcal{R}_{[0,r]}(I), kr) \subseteq \Phi(P_1, kr) = P_{k+1} \subseteq R_N.$$

D'où $y \in R_N$.

Calculons maintenant l'erreur produite par CheckMate. On a

$$R_N = \bigcup_{k=0}^{k=N-1} \Phi(\hat{\mathcal{R}}_{[0,r]}(I), kr).$$

Soit $x \in R_N$,

$$\exists k \in \{0, \dots, N-1\}, y \in \hat{\mathcal{R}}_{[0,r]}(I), x = \Phi(y, kr).$$

De plus,

$$d_H(\mathcal{R}_{[0,r]}(I), \hat{\mathcal{R}}_{[0,r]}(I)) = \delta_r(I) \implies \exists y' \in \mathcal{R}_{[0,r]}(I), \|y - y'\| \leq \delta_r(I).$$

Or, $x' = \Phi(y', kr)$ est un élément de $\mathcal{R}_{[0,t]}(I)$ et

$$x' - x = \Phi(y', kr) - \Phi(y, kr) = e^{Akr}(y' - y).$$

Donc,

$$\|x' - x\| \leq \|e^{Akr}\| \|y' - y\| \leq e^{\|A\|(t-r)} \delta_r(I).$$

■

6.2.5 Comparaison des deux méthodes

Nous avons vu que le calcul de l'image d'un polytope par le flot pouvait être effectué facilement et de manière précise. Par conséquent, une bonne idée de la complexité de chaque algorithme est donnée par le nombre d'appels à la fonction $\hat{\mathcal{R}}_{[0,r]}$. De ce point de vue, il est clair que Checkmate est meilleur que d/dt. Il n'appelle la fonction $\hat{\mathcal{R}}_{[0,r]}$ qu'une seule fois (lors de l'initialisation) alors que d/dt l'appelle N fois (un appel à chaque itération).

De plus, l'erreur d'approximation commise par les deux méthodes est comparable (de l'ordre de $O(r)$). La méthode utilisée par CheckMate paraît donc, en théorie, plus efficace que celle de d/dt.

Remarque 6.2.1 *L'ordre de convergence des méthodes est en $O(r)$, cependant, en pratique et pour les raisons déjà invoquées, il arrive fréquemment que celui-ci soit en $O(r^2)$.*

6.3 Atteignabilité des systèmes linéaires soumis à des perturbations

On considère maintenant un système dynamique linéaire soumis à des perturbations

$$x'(t) = Ax(t) + b + u(t), \quad x(t) \in \mathbb{R}^n, \quad \|u(t)\| \leq \mu.$$

On souhaite calculer une approximation de l'ensemble atteignable du système pour l'ensemble de conditions initiales I .

Notons $\Phi^\mu(y, t)$, l'ensemble des points de \mathbb{R}^n atteignables à l'instant t pour la condition initiale y . x est un élément de $\Phi^\mu(y, t)$ si et seulement si il existe une fonction entrée u , mesurable, telle que $\|u(s)\| \leq \mu$ presque partout sur $[0, t]$ (on dit que u est admissible) et

$$x = e^{At}y + \int_0^t e^{A(t-s)}(b + u(s)) ds.$$

On notera, pour un ensemble $I \subseteq \mathbb{R}^n$,

$$\Phi^\mu(I, t) = \bigcup_{y \in I} \Phi^\mu(y, t).$$

Définition 6.3.1 *L'ensemble atteignable du système dynamique sur l'intervalle $[0, t]$ pour un ensemble de conditions initiales I , noté $\mathcal{R}_{[0,t]}^\mu(I)$, est donné par :*

$$\mathcal{R}_{[0,t]}^\mu(I) = \bigcup_{s \in [0,t]} \Phi^\mu(I, s).$$

Comme pour les systèmes autonomes, on cherche à approcher cet ensemble de manière conservative. Dans [24], une généralisation de l'algorithme 4, basée sur le principe de maximum de Pontryagin [71], a été proposée. Dans cette partie, nous nous proposons d'adapter l'algorithme 5 pour traiter les systèmes linéaires soumis à des perturbations².

Tout d'abord, observons que la décomposition de l'ensemble atteignable donnée par l'équation (6.2.3) peut être généralisée aux systèmes avec entrée :

Proposition 6.3.1 *Soient $r > 0$, $N \in \mathbb{N}$, on note $t = Nr$,*

$$\mathcal{R}_{[0,t]}^\mu(I) = \bigcup_{k=0}^{k=N-1} \Phi^\mu(\mathcal{R}_{[0,r]}^\mu(I), kr).$$

Preuve : On a clairement

$$\mathcal{R}_{[0,t]}^\mu(I) = \bigcup_{k=0}^{k=N-1} \mathcal{R}_{[kr, (k+1)r]}^\mu(I).$$

²Travail mené en collaboration avec E. Asarin et T. Dang

Soit x un élément de $\mathcal{R}_{[kr, (k+1)r]}^\mu(I)$, il existe $y \in I$, $s \in [kr, (k+1)r]$, et u une fonction entrée admissible, tels que

$$\begin{aligned} x &= e^{As}y + \int_0^s e^{A(s-\tau)}(b + u(\tau)) d\tau. \\ &= e^{As}y + \int_0^{s-kr} e^{A(s-\tau)}(b + u(\tau)) d\tau + \int_{s-kr}^s e^{A(s-\tau)}(b + u(\tau)) d\tau. \\ &= e^{Akr} \left(e^{A(s-kr)}y + \int_0^{s-kr} e^{A(s-kr-\tau)}(b + u(\tau)) d\tau \right) \\ &\quad + \int_0^{kr} e^{A(kr-\tau)}(b + u(\tau + s - kr)) d\tau. \end{aligned}$$

En remarquant que

$$z = e^{A(s-kr)}y + \int_0^{s-kr} e^{A(s-kr-\tau)}(b + u(\tau)) d\tau$$

est un élément de $\mathcal{R}_{[0,r]}^\mu(I)$, on peut déduire que

$$x = e^{Akr}z + \int_0^{kr} e^{A(kr-\tau)}(b + u(\tau + s - kr)) d\tau$$

est un élément de $\Phi^\mu(\mathcal{R}_{[0,r]}^\mu(I), kr)$. Réciproquement, par le cheminement inverse, il est assez simple de voir qu'un élément de $\Phi^\mu(\mathcal{R}_{[0,r]}^\mu(I), kr)$ est un élément de $\mathcal{R}_{[kr, (k+1)r]}^\mu(I)$. ■

Ainsi, en adaptant l'algorithme 5, une approximation conservative de l'ensemble atteignable peut être calculée de la manière suivante :

1. On calcule un polytope $\hat{\mathcal{R}}_{[0,r]}^\mu(I)$, approximation conservative de l'ensemble atteignable $\mathcal{R}_{[0,r]}^\mu(I)$.
2. On calcule les images $\Phi^\mu(\hat{\mathcal{R}}_{[0,r]}^\mu(I), kr)$ de cet ensemble par le flot.

Contrairement au cas autonome, l'image d'un polytope I par le flot Φ^μ n'est pas simple à calculer. Il faut donc, là aussi, avoir recours à une méthode d'approximation. On doit donc définir une fonction $\hat{\Phi}^\mu$, qui à un polytope I et à un réel positif r , associe un polytope $\hat{\Phi}^\mu(I, r)$ qui approche l'ensemble $\Phi^\mu(I, r)$ de manière conservative.

6.3.1 Approximation conservative de l'image par le flot

La méthode d'approximation conservative de l'image d'un polytope par le flot repose sur le résultat suivant :

Lemme 6.3.1 *Pour tout $y \in \mathbb{R}^n$, $r \geq 0$,*

$$\Phi^\mu(y, r) \subseteq \mathcal{B} \left(\Phi(y, r), \frac{\mu}{\|A\|} (e^{\|A\|r} - 1) \right)$$

où Φ dénote le flot de l'équation différentielle linéaire sans le terme de perturbation.

Preuve : Soit $x \in \Phi^\mu(y, r)$, il existe une fonction entrée u admissible telle que

$$\begin{aligned} x &= e^{Ar}y + \int_0^r e^{A(r-s)}(b + u(s)) ds \\ &= e^{Ar}y + \int_0^r e^{A(r-s)}b ds + \int_0^r e^{A(r-s)}u(s) ds \\ &= \Phi(y, r) + \int_0^r e^{A(r-s)}u(s) ds. \end{aligned}$$

Donc,

$$\|x - \Phi(y, r)\| \leq \int_0^r e^{\|A\|(r-s)}\|u(s)\| ds \leq \mu \int_0^r e^{\|A\|(r-s)} ds \leq \frac{\mu}{\|A\|}(e^{\|A\|r} - 1).$$

■

L'ensemble des points, que l'on peut atteindre à l'instant r , en partant d'un point $y \in \mathbb{R}^n$, sont donc à une distance inférieure à $\frac{\mu}{\|A\|}(e^{\|A\|r} - 1)$ de l'image de y par le flot de l'équation différentielle autonome. Par conséquent, l'ensemble $\hat{\Phi}^\mu(I, r)$, donné par la relation suivante, nous donne une approximation conservative de $\Phi^\mu(I, r)$:

$$\hat{\Phi}^\mu(I, r) = \Phi(I, r) \oplus \mathcal{B}\left(0, \frac{\mu}{\|A\|}(e^{\|A\|r} - 1)\right). \quad (6.3.4)$$

De plus, la proposition suivante garantit que cette approximation est de bonne qualité.

Proposition 6.3.2 *Soit $\hat{\Phi}^\mu(I, r)$, l'ensemble défini par l'équation (6.3.4), $\hat{\Phi}^\mu(I, r)$ est un polytope et*

$$\Phi^\mu(I, r) \subseteq \hat{\Phi}^\mu(I, r).$$

De plus,

$$d_H(\Phi^\mu(I, r), \hat{\Phi}^\mu(I, r)) \leq \mu\|A\|e^{\|A\|r}r^2.$$

Preuve : D'après la proposition 6.2.2, $\Phi(I, r)$ est un polytope. De plus, la norme $\|\cdot\|$ est polyédrique, par conséquent $\hat{\Phi}^\mu(I, r)$ est un polytope.

D'après le lemme 6.3.1, il est clair que $\Phi^\mu(I, r) \subseteq \hat{\Phi}^\mu(I, r)$.

Il reste donc à évaluer la qualité de l'approximation. Soit $x \in \hat{\Phi}^\mu(I, r)$, il existe $y \in \Phi(I, r)$ tel que

$$\|x - y\| \leq \frac{\mu}{\|A\|}(e^{\|A\|r} - 1).$$

Il existe également $z \in I$ tel que $y = \Phi(z, r)$.

Définissons la fonction entrée u , constante sur $[0, r]$:

$$\forall s \in [0, r], u(s) = \frac{\|A\|}{e^{\|A\|r} - 1}(x - y).$$

Pour tout $s \in [0, r]$, $\|u(s)\| \leq \mu$ et donc u est une entrée admissible du système dynamique. Soit alors x_u , le point de $\Phi^\mu(I, r)$ atteint à l'instant r pour la condition initiale z et l'entrée u . On a

$$\begin{aligned} x_u &= e^{Ar}z + \int_0^r e^{A(r-s)}(b + u(s)) ds = \Phi(z, r) + \int_0^r e^{A(r-s)}u(s) ds \\ &= y + \frac{\|A\|}{e^{\|A\|r} - 1} \int_0^r e^{A(r-s)}(x - y) ds. \end{aligned}$$

D'où,

$$\begin{aligned}
 x_u - x &= y - x + \frac{\|A\|}{e^{\|A\|r} - 1} \int_0^r e^{A(r-s)}(x - y) ds \\
 &= \frac{\|A\|}{e^{\|A\|r} - 1} \int_0^r e^{A(r-s)}(x - y) - \frac{e^{\|A\|r} - 1}{\|A\|r}(x - y) ds \\
 &= \frac{\|A\|}{e^{\|A\|r} - 1} \int_0^r (e^{A(r-s)} - I)(x - y) - \frac{e^{\|A\|r} - \|A\|r - 1}{\|A\|r}(x - y) ds.
 \end{aligned}$$

Donc,

$$\begin{aligned}
 \|x_u - x\| &\leq \frac{\|A\|}{e^{\|A\|r} - 1} \int_0^r \|e^{A(r-s)} - I\| \|x - y\| + \frac{e^{\|A\|r} - \|A\|r - 1}{\|A\|r} \|x - y\| ds \\
 &\leq \frac{\|A\|}{e^{\|A\|r} - 1} \|x - y\| \int_0^r (r - s) \|A\| e^{\|A\|(r-s)} + \frac{r\|A\|e^{\|A\|r}}{2} ds \\
 &\leq \mu r^2 \|A\| e^{\|A\|r}.
 \end{aligned}$$

■

On a donc la possibilité d'approcher précisément (en $O(r^2\mu)$) l'ensemble $\Phi^\mu(I, r)$ par un polytope (voir figure 6.4).

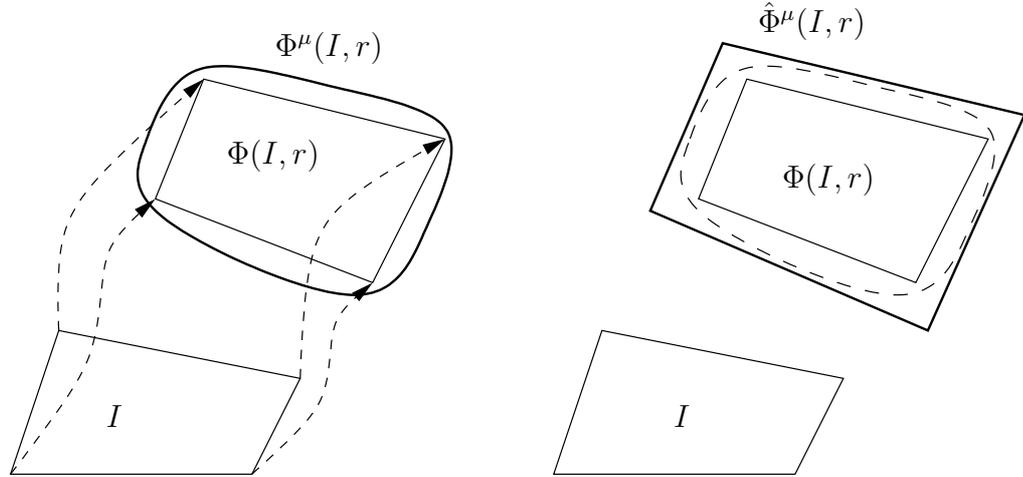


FIG. 6.4: Illustration de la méthode de calcul de l'ensemble $\hat{\Phi}^\mu(I, r)$

Il nous reste à calculer une approximation conservative $\hat{\mathcal{R}}_{[0,r]}^\mu(I)$ de l'ensemble atteignable $\mathcal{R}_{[0,r]}^\mu(I)$.

6.3.2 Approximation conservative de $\hat{\mathcal{R}}_{[0,r]}^\mu(I)$

Soit I un polytope, on souhaite approcher de manière conservative, l'ensemble atteignable $\mathcal{R}_{[0,r]}^\mu(I)$ par un polytope $\hat{\mathcal{R}}_{[0,r]}^\mu(I)$.

L'idée du calcul est la suivante. Tout d'abord, on calcule, par l'algorithme 3, $\hat{\mathcal{R}}_{[0,r]}(I)$, l'approximation conservatrice de $\mathcal{R}_{[0,r]}(I)$, l'ensemble atteignable du système dynamique autonome. On prend ensuite, un voisinage de $\hat{\mathcal{R}}_{[0,r]}(I)$, suffisamment grand pour contenir l'ensemble $\mathcal{R}_{[0,r]}^\mu(I)$.

Soit $x \in \mathcal{R}_{[0,r]}^\mu(I)$, il existe $s \in [0, r]$ et $y \in I$ tels que $x \in \Phi^\mu(y, s)$. D'après le lemme 6.3.1, on sait que

$$\|x - \Phi(y, s)\| \leq \frac{\mu}{\|A\|} (e^{\|A\|s} - 1) \leq \frac{\mu}{\|A\|} (e^{\|A\|r} - 1).$$

Donc, pour tout $x \in \mathcal{R}_{[0,r]}^\mu(I)$, il existe un point de $\mathcal{R}_{[0,r]}(I)$ à une distance inférieure à $\frac{\mu}{\|A\|} (e^{\|A\|r} - 1)$. D'où,

$$\mathcal{R}_{[0,r]}^\mu(I) \subseteq \mathcal{R}_{[0,r]}(I) \oplus \mathcal{B}\left(0, \frac{\mu}{\|A\|} (e^{\|A\|r} - 1)\right). \quad (6.3.5)$$

Une approximation conservatrice de $\mathcal{R}_{[0,r]}^\mu(I)$ par un polytope est donc donnée par

$$\hat{\mathcal{R}}_{[0,r]}^\mu(I) = \hat{\mathcal{R}}_{[0,r]}(I) \oplus \mathcal{B}\left(0, \frac{\mu}{\|A\|} (e^{\|A\|r} - 1)\right) \quad (6.3.6)$$

où $\hat{\mathcal{R}}_{[0,r]}(I)$ est l'ensemble calculé par l'algorithme 3.

Proposition 6.3.3 Soit $\hat{\mathcal{R}}_{[0,r]}^\mu(I)$ l'ensemble défini par l'équation (6.3.6), $\hat{\mathcal{R}}_{[0,r]}^\mu(I)$ est un polytope et

$$\mathcal{R}_{[0,r]}^\mu(I) \subseteq \hat{\mathcal{R}}_{[0,r]}^\mu(I).$$

De plus,

$$d_H(\mathcal{R}_{[0,r]}^\mu(I), \hat{\mathcal{R}}_{[0,r]}^\mu(I)) \leq \delta_r(I) + \mu r e^{\|A\|r}$$

où $\delta_r(I)$ est donné par la proposition 6.2.3.

Preuve : D'après la proposition 6.2.3, $\hat{\mathcal{R}}_{[0,r]}(I)$ est un polytope, donc $\hat{\mathcal{R}}_{[0,r]}^\mu(I)$ est également un polytope.

D'après les équations (6.3.5) et (6.3.6), il est clair que $\mathcal{R}_{[0,r]}^\mu(I) \subseteq \hat{\mathcal{R}}_{[0,r]}^\mu(I)$.

Soit $x \in \hat{\mathcal{R}}_{[0,r]}^\mu(I)$, il existe $y \in \hat{\mathcal{R}}_{[0,r]}(I)$ tel que

$$\|x - y\| \leq \frac{\mu}{\|A\|} (e^{\|A\|r} - 1) \leq \mu r e^{\|A\|r}.$$

D'après la proposition 6.2.3, il existe $z \in \mathcal{R}_{[0,r]}(I)$, tel que $\|y - z\| \leq \delta_r(I)$. Donc,

$$\|x - z\| \leq \delta_r(I) + \mu r e^{\|A\|r}.$$

De plus, $\mathcal{R}_{[0,r]}(I) \subseteq \mathcal{R}_{[0,r]}^\mu(I)$, donc, z est également un élément de $\mathcal{R}_{[0,r]}^\mu(I)$. ■

On a donc une méthode pour approcher l'ensemble $\mathcal{R}_{[0,r]}^\mu(I)$ par un polytope $\hat{\mathcal{R}}_{[0,r]}^\mu(I, r)$ (voir figure 6.5).

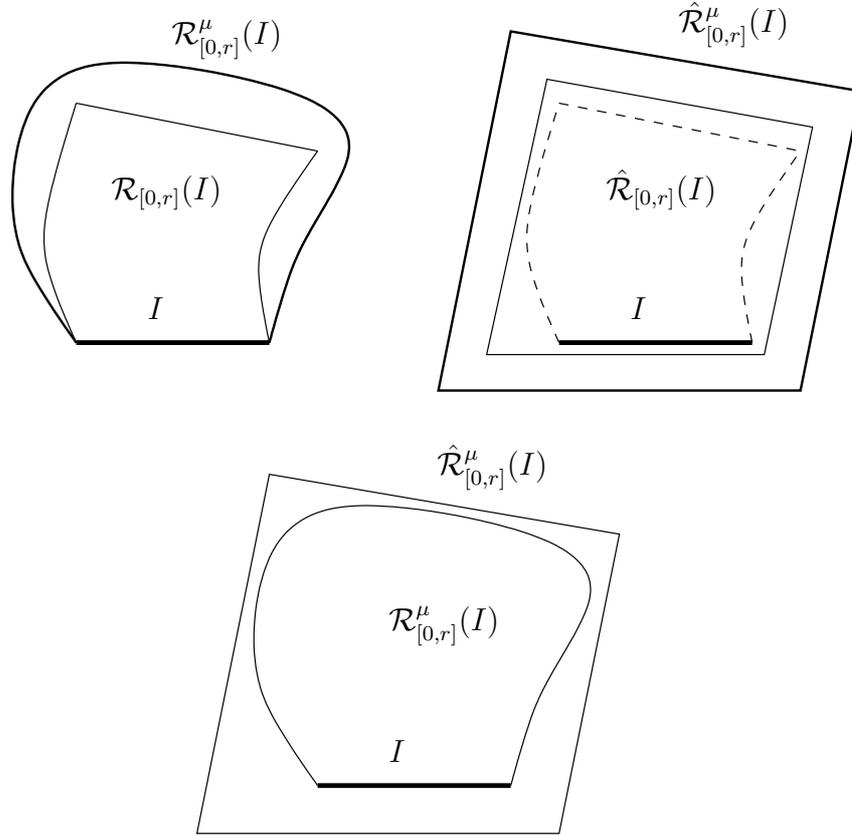


FIG. 6.5: Illustration de la méthode de calcul de l'ensemble $\hat{\mathcal{R}}_{[0,r]}^\mu(I, r)$

L'approximation est en $O(\mu r + \delta_r(I))$. Par rapport au cas autonome, on perd donc le caractère quadratique de la convergence que l'on avait dans certains cas. Cependant, il convient de noter que comme le terme $u(t)$ représente des perturbations, il est raisonnable de le considérer comme petit. Par conséquent, en faisant l'hypothèse que μ est relativement petit, l'approximation reste de qualité tout à fait correcte.

6.3.3 Approximation conservative de l'ensemble atteignable

À présent, nous avons, à notre disposition, les outils nécessaires pour mettre en oeuvre la généralisation de l'algorithme 5 aux systèmes linéaires soumis à des perturbations :

- On initialise le calcul de l'ensemble atteignable en calculant l'approximation conservative $\hat{\mathcal{R}}_{[0,r]}^\mu(I)$ de l'ensemble $\mathcal{R}_{[0,r]}^\mu(I)$.
- On propage cet ensemble grâce à l'approximation conservative $\hat{\Phi}^\mu$ du flot Φ^μ du système dynamique soumis à des perturbations.

L'algorithme 6 décrit la méthode de calcul de l'approximation conservative de l'ensemble atteignable $\mathcal{R}_{[0,t]}^\mu(I)$.

Données : Un polytope I .

Résultat : Une approximation conservative R_N de l'ensemble $\mathcal{R}_{[0,t]}^\mu(I)$.

$$\varepsilon := \frac{\mu}{\|A\|} (e^{\|A\|r} - 1);$$

– *Initialisation*

$$P_1 := \hat{\mathcal{R}}_{[0,r]}^\mu(I);$$

$$Q_1 := P_1 \oplus \mathcal{B}(0, \varepsilon);$$

$$R_1 := Q_1;$$

– *Boucle principale*

pour $k = 1 \dots N - 1$ **faire**

$$\left| \begin{array}{l} P_{k+1} := \Phi(Q_k, r); \\ Q_{k+1} := P_{k+1} \oplus \mathcal{B}(0, \varepsilon); \\ R_{k+1} := R_k \cup Q_{k+1}; \end{array} \right.$$

fin

Algorithme 6: Calcul de l'ensemble atteignable

Q_{k+1} est une approximation conservative de $\mathcal{R}_{[kr, (k+1)r]}^\mu(I)$, l'ensemble atteignable du système dynamique sur l'intervalle $[kr, (k+1)r]$. À l'initialisation, on affecte à Q_1 l'ensemble $\hat{\mathcal{R}}_{[0,r]}^\mu(I)$. Le calcul se fait en deux temps (P_1 est une variable intermédiaire). Ensuite, à chaque itération de la boucle, on affecte à Q_{k+1} l'image de l'ensemble Q_k par l'approximation conservative du flot

$$Q_{k+1} = \hat{\Phi}^\mu(Q_k, r).$$

Là encore, le calcul se fait en deux temps (P_{k+1} est une variable intermédiaire).

R_{k+1} est une approximation conservative de l'ensemble atteignable du système dynamique sur l'intervalle $[0, (k+1)r]$. On a

$$R_{k+1} = \bigcup_{i=1}^{i=k+1} Q_i.$$

L'ensemble R_N obtenu à la sortie de l'algorithme est une approximation conservative de $\mathcal{R}_{[0,t]}^\mu(I)$.

Remarque 6.3.1 *Lors de l'exécution de l'algorithme 6, on effectue de nombreuses sommes de Minkowski. À chaque itération une opération de ce type est calculée. Le nombre de sommets de l'ensemble Q_{k+1} peut alors augmenter exponentiellement.*

Ceci pose des problèmes de gestion de mémoire et de complexité de l'algorithme. De plus, les sommets ainsi introduits étant souvent très rapprochés, cela peut poser des problèmes de stabilité numérique.

Pour toutes ces raisons, il est souvent judicieux d'approcher la somme de Minkowski de manière conservative (voir figure 6.6). On perd en précision de calcul, mais on gagne énormément en temps de calcul et en espace mémoire utilisé.

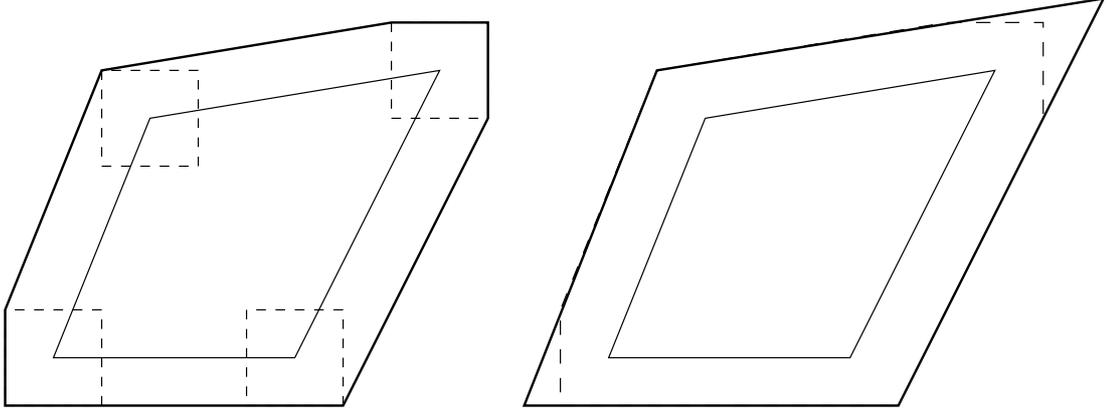


FIG. 6.6: Approximation conservative de la somme de Minkowski. Le nombre de sommets reste constant.

Le théorème suivant suppose que les sommes de Minkowski sont calculées de manière exacte.

Théorème 6.3.1 Soit R_N l'ensemble calculé par l'algorithme 6 :

1. (Approximation conservative) $\mathcal{R}_{[0,t]}^\mu(I) \subseteq R_N$.
2. (Convergence de l'approximation)

$$d_H(\mathcal{R}_{[0,t]}^\mu(I), R_N) \leq (\delta_r(I)e^{-\|A\|r} + 2\mu r)e^{\|A\|t}$$

où $\delta_r(I)$ est donné par la proposition 6.2.3.

Preuve : $Q_1 = \hat{\mathcal{R}}_{[0,r]}^\mu(I)$, par conséquent, il est clair que

$$\mathcal{R}_{[0,r]}^\mu(I) \subseteq Q_1.$$

Soit $k \in \{1, \dots, N-1\}$, supposons que,

$$\mathcal{R}_{[(k-1)r, kr]}^\mu(I) \subseteq Q_k.$$

Alors

$$\Phi^\mu(\mathcal{R}_{[(k-1)r, kr]}^\mu(I), r) \subseteq \Phi^\mu(Q_k, r) \subseteq \hat{\Phi}^\mu(Q_k, r).$$

En remarquant que

$$\Phi^\mu(\mathcal{R}_{[(k-1)r, kr]}^\mu(I), r) = \mathcal{R}_{[kr, (k+1)r]}^\mu(I) \text{ et } \hat{\Phi}^\mu(Q_k, r) = Q_{k+1},$$

on arrive à

$$\mathcal{R}_{[kr, (k+1)r]}^\mu(I) \subseteq Q_{k+1}.$$

L'hypothèse de récurrence étant vraie pour $k=1$, elle est vraie pour tout $k \in \{1, \dots, N\}$.

En utilisant la décomposition de l'ensemble atteignable, donnée par la proposition 6.3.1, on arrive à

$$\mathcal{R}_{[0,t]}^\mu(I) = \bigcup_{k=1}^{k=N} \mathcal{R}_{[(k-1)r, kr]}^\mu(I) \subseteq \bigcup_{k=1}^{k=N} Q_k = R_N.$$

Démontrons maintenant la propriété de convergence. D'après la deuxième assertion de la proposition 6.1.1,

$$d_H(\mathcal{R}_{[0,t]}^\mu(I), R_N) \leq \max_{k=0}^{k=N-1} d_H(\mathcal{R}_{[kr, (k+1)r]}^\mu(I), Q_{k+1}). \quad (6.3.7)$$

D'après la proposition 6.3.3,

$$d_H(\mathcal{R}_{[0,r]}^\mu(I), Q_1) = d_H(\mathcal{R}_{[0,r]}^\mu(I), \hat{\mathcal{R}}_{[0,r]}^\mu(I)) \leq \delta_r(I) + \mu r e^{\|A\|r}.$$

Soit $k \in \{1, \dots, N-1\}$,

$$\begin{aligned} d_H(\mathcal{R}_{[kr, (k+1)r]}^\mu(I), Q_{k+1}) &= d_H\left(\Phi^\mu(\mathcal{R}_{[(k-1)r, kr]}^\mu(I), r), \hat{\Phi}^\mu(Q_k, r)\right) \\ &\leq d_H\left(\Phi^\mu(\mathcal{R}_{[(k-1)r, kr]}^\mu(I), r), \Phi^\mu(Q_k, r)\right) \\ &\quad + d_H\left(\Phi^\mu(Q_k, r), \hat{\Phi}^\mu(Q_k, r)\right). \end{aligned} \quad (6.3.8)$$

D'après la proposition 6.3.2, on a

$$d_H\left(\Phi^\mu(Q_k, r), \hat{\Phi}^\mu(Q_k, r)\right) \leq \mu \|A\| e^{\|A\|r} r^2. \quad (6.3.9)$$

Soit $x \in \Phi^\mu(Q_k, r)$, il existe $y \in Q_k$ et une fonction entrée admissible u tels que

$$x = e^{Ar} y + \int_0^r e^{A(r-s)} (b + u(s)) ds.$$

Il existe également $z \in \mathcal{R}_{[(k-1)r, kr]}^\mu(I)$, tel que

$$\|y - z\| \leq d_H(\mathcal{R}_{[(k-1)r, kr]}^\mu(I), Q_k).$$

Soit x' l'élément de $\Phi^\mu(\mathcal{R}_{[(k-1)r, kr]}^\mu(I), r)$ défini par

$$x' = e^{Ar} z + \int_0^r e^{A(r-s)} (b + u(s)) ds.$$

On a

$$\|x - x'\| = \|e^{Ar}(y - z)\| \leq e^{\|A\|r} \|y - z\| \leq e^{\|A\|r} d_H(\mathcal{R}_{[(k-1)r, kr]}^\mu(I), Q_k).$$

De plus, en remarquant que $\Phi^\mu(\mathcal{R}_{[(k-1)r, kr]}^\mu(I), r) \subseteq \Phi^\mu(Q_k, r)$, on arrive à

$$d_H\left(\Phi^\mu(\mathcal{R}_{[(k-1)r, kr]}^\mu(I), r), \Phi^\mu(Q_k, r)\right) \leq e^{\|A\|r} d_H(\mathcal{R}_{[(k-1)r, kr]}^\mu(I), Q_k). \quad (6.3.10)$$

En réinjectant les équations (6.3.9) et (6.3.10) dans l'équation (6.3.8), on arrive à

$$d_H(\mathcal{R}_{[kr, (k+1)r]}^\mu(I), Q_{k+1}) \leq e^{\|A\|r} d_H(\mathcal{R}_{[(k-1)r, kr]}^\mu(I), Q_k) + \mu \|A\| e^{\|A\|r} r^2.$$

D'où,

$$\begin{aligned} d_H(\mathcal{R}_{[kr, (k+1)r]}^\mu(I), Q_{k+1}) &\leq e^{k\|A\|r} d_H(\mathcal{R}_{[0,r]}^\mu(I), Q_1) + \mu \|A\| e^{\|A\|r} r^2 \sum_{i=0}^{i=k-1} e^{i\|A\|r} \\ &\leq e^{k\|A\|r} \left(\delta_r(I) + \mu r e^{\|A\|r} \right) + \mu \|A\| e^{\|A\|r} r^2 \frac{e^{\|A\|kr} - 1}{e^{\|A\|r} - 1}. \end{aligned}$$

En remarquant que $e^{\|A\|r} - 1 \geq \|A\|r$, on arrive à

$$d_H(\mathcal{R}_{[kr, (k+1)r]}^\mu(I), Q_{k+1}) \leq e^{k\|A\|r} \delta_r(I) + 2\mu r e^{\|A\|(k+1)r}.$$

L'équation (6.3.7) permet de conclure. ■

Dans ce chapitre, nous avons donc proposé une généralisation de l'algorithme 5, utilisé par la boîte à outils CheckMate, pour le calcul de l'ensemble atteignable d'un système dynamique linéaire autonome. L'algorithme 6 résultant permet de calculer une approximation conservative de l'ensemble atteignable d'un système dynamique linéaire soumis à des perturbations. Cette approximation, représentée sous forme d'union de polytopes, converge vers l'ensemble atteignable quand le nombre de polytopes tend vers l'infini.

6.3.4 Exemple

On considère le système dynamique linéaire suivant :

$$x'(t) = Ax(t) + b + u(t), \|u(t)\| \leq \mu$$

où

$$A = \begin{pmatrix} -1 & 0 \\ 1 & -1 \end{pmatrix}, b = \begin{pmatrix} -2 \\ -5 \end{pmatrix}.$$

Ce système dynamique correspond au troisième mode du système de deux réservoirs présenté dans le chapitre précédent. On souhaite calculer l'ensemble atteignable du système dynamique pour l'ensemble de conditions initiales :

$$I = \left\{ (x_1, x_2) \in \mathbb{R}^2, \frac{3}{2} \leq x_1 \leq \frac{5}{2}, x_2 = 1 \right\}.$$

La figure 6.7 représente l'évolution de l'ensemble atteignable du système dynamique autonome ($\mu = 0$). Les ensembles ont été calculés par l'algorithme 5 avec un pas de temps $r = 0.1$. Le dernier ensemble calculé, en trait plein sur la figure, est une approximation conservative de l'ensemble atteignable $\mathcal{R}_{[0,1]}(I)$.

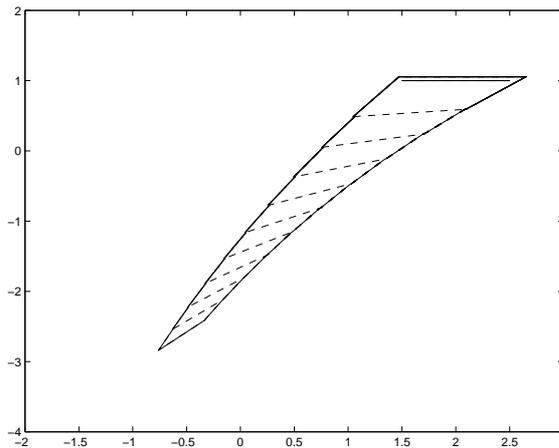


FIG. 6.7: Evolution de l'ensemble atteignable du système dynamique autonome $\mathcal{R}_{[0,t]}(I)$. L'ensemble représenté en trait plein est une approximation conservative de l'ensemble atteignable $\mathcal{R}_{[0,1]}(I)$.

Sur la figure 6.8 nous avons représenté l'évolution de l'ensemble atteignable du système dynamique soumis à des perturbations pour des valeurs de μ respectivement égales à 0.2 et 0.5. Les ensembles ont été calculés par l'algorithme 6 avec un pas de temps $r = 0.1$. Le dernier ensemble calculé, en trait plein sur la figure, est une approximation conservative de l'ensemble atteignable $\mathcal{R}_{[0,1]}^\mu(I)$.

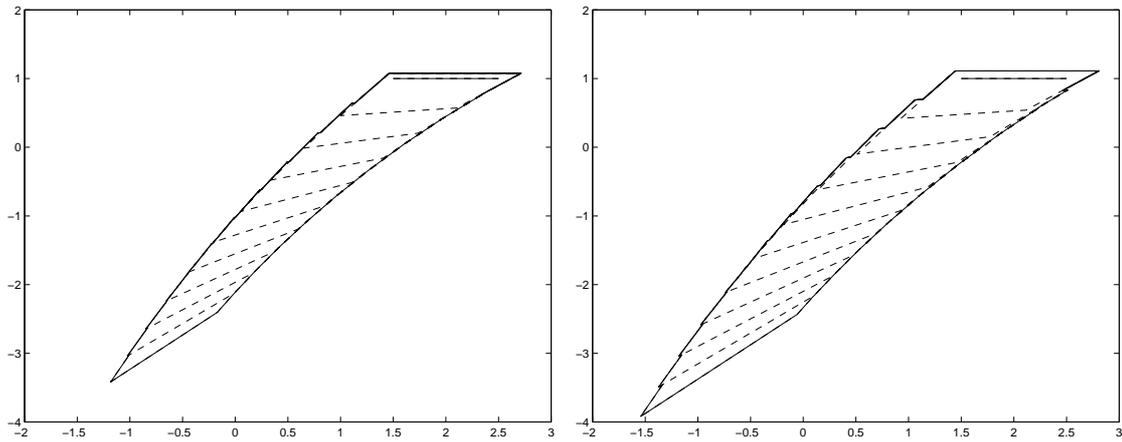


FIG. 6.8: Evolution de l'ensemble atteignable du système dynamique soumis à des perturbations $\mathcal{R}_{[0,t]}^\mu(I)$ ($\mu = 0.2$ à gauche, $\mu = 0.5$ à droite). L'ensemble représenté en trait plein est une approximation conservative de l'ensemble atteignable $\mathcal{R}_{[0,1]}^\mu(I)$.

Sur la figure 6.9, nous avons tracé les ensembles atteignables $\mathcal{R}_{[0,1]}^\mu(I)$ du système dynamique pour différentes valeurs de μ . On peut observer sur cette figure la croissance de l'ensemble atteignable en fonction du paramètre μ qui borne la norme des perturbations tolérées par le système.

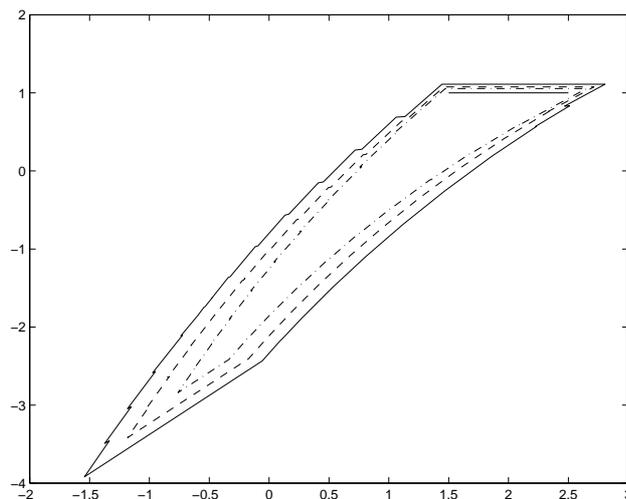


FIG. 6.9: Approximations conservatives des ensembles atteignables $\mathcal{R}_{[0,1]}^\mu(I)$ (de l'intérieur vers l'extérieur, $\mu = 0$, $\mu = 0.2$, $\mu = 0.5$).

Chapitre 7

Atteignabilité des Systèmes Hybrides

Dans ce chapitre, on considère un système hybride linéaire par morceaux \mathcal{H} , tel que dans chaque mode discret, la dynamique continue du système est donnée par un système dynamique soumis à des perturbations, similaire à ceux traités dans le chapitre précédent.

7.1 Ensemble atteignable d'un système hybride

On se donne un ensemble de conditions initiales $I \subseteq \mathcal{Q} \times \mathbb{R}^n$. Q_0 dénote l'ensemble des états discrets initiaux. Pour tout $q \in Q_0$, $I_0^q \subseteq D_q$ est l'ensemble des valeurs initiales possibles pour la variable continue, si la valeur initiale de l'état discret est q . On a donc :

$$I = \bigcup_{q \in Q_0} (\{q\} \times I_0^q).$$

Dans l'idéal, on souhaiterait calculer $\mathcal{R}(I)$, l'ensemble atteignable par le système hybride. Un couple $(\sigma, y) \in \mathcal{Q} \times \mathbb{R}^n$ est dans l'ensemble $\mathcal{R}(I)$, si il existe une exécution (τ, q, x) , acceptée par le système hybride et telle que

$$(q(t_0), x(t_0)) \in I, \text{ et } \exists t \in \tau, q(t) = \sigma, x(t) = y. \quad (7.1.1)$$

Malheureusement, il est rare que cet ensemble, ou bien une bonne approximation, soit calculable. Par exemple, il est possible que le nombre de transitions ou bien la durée d'exécution, nécessaires pour atteindre certains états, ne soient pas bornés. Dans ce cas, le calcul de l'ensemble $\mathcal{R}(I)$ ne peut, en général, pas se faire en temps fini (on doit simuler une infinité de transitions ou bien une durée infinie). Nous définissons donc une notion plus faible de l'ensemble atteignable.

Définition 7.1.1 *L'ensemble atteignable $\mathcal{R}_{p,s}(I)$ du système hybride est l'ensemble des couples $(\sigma, y) \in \mathcal{Q} \times \mathbb{R}^n$ tels qu'il existe une exécution (τ, q, x) , acceptée par le système hybride, et vérifiant,*

$$\begin{cases} \tau = \{[t_i, t_{i+1}]\}_{i=0}^{i=N}, \text{ avec } N \leq p, \text{ et } \forall i \in \{0, \dots, N\}, t_{i+1} - t_i \leq s \\ (q(t_0), x(t_0)) \in I, \text{ et } \exists t \in [t_N, t_{N+1}], q(t) = \sigma, x(t) = y. \end{cases}$$

Autrement dit, l'ensemble atteignable $\mathcal{R}_{p,s}(I)$ est l'ensemble des états que l'on peut atteindre en moins de p transitions, et en restant une durée inférieure à s dans chaque mode.

Lemme 7.1.1

$$\mathcal{R}(I) = \bigcup_{p \in \mathbb{N}, s \in \mathbb{R}^+} \mathcal{R}_{p,s}(I).$$

Preuve : Il est clair que, pour tout $p \in \mathbb{N}$, pour tout $s \geq 0$, $\mathcal{R}_{p,s}(I) \subseteq \mathcal{R}(I)$.

Soit $(\sigma, y) \in \mathcal{R}(I)$, il existe une exécution (τ, q, x) vérifiant (7.1.1). On note p l'indice de l'élément de $\tau = \{[t_i, t_{i+1}]\}_{i=0}^{i=N}$ contenant l'instant t auquel $q(t) = \sigma$ et $x(t) = y$. Soit s un réel positif vérifiant

$$\forall i \in \{0, \dots, p-1\}, t_{i+1} - t_i \leq s \text{ et } t - t_p \leq s.$$

On a clairement, $(\sigma, y) \in \mathcal{R}_{p,s}(I)$. ■

Dans ce chapitre, nous proposons un algorithme pour calculer une approximation conservative de l'ensemble $\mathcal{R}_{p,s}(I)$. Dans certains cas, que nous identifierons, l'ensemble calculé fournira également une approximation conservative de l'ensemble $\mathcal{R}(I)$.

Globalement, l'algorithme de calcul de l'ensemble atteignable consiste en une boucle principale. A la k -ième itération de la boucle, on calcule les états atteignables en $k-1$ transitions à partir de l'ensemble de conditions initiales I .

Notons Q_k , l'ensemble des états discrets atteignables en k transitions. Pour $q \in Q_k$, on note I_k^q , l'ensemble des valeurs de D_q que la variable continue du système hybride peut valoir lors de la transition vers q .

Chaque itération se déroule en deux phases.

1. Calcul des successeurs continus :

Cette phase consiste à calculer, pour chaque élément q de Q_k , $\mathcal{X}_s^q(I_k^q)$ le sous-ensemble de D_q , atteignable par la dynamique continue associée à l'état q , sur l'intervalle de temps $[0, s]$, à partir de l'ensemble I_k^q .

2. Calcul des successeurs discrets :

On regarde quelles gardes G_e ($e = (q, q')$) ont été atteintes :

$$\mathcal{X}_s^q(I_k^q) \cap G_e \neq \emptyset.$$

L'état q' est ajouté à Q_{k+1} . L'ensemble $\mathcal{X}_s^q(I_k^q) \cap G_e$ est ajouté à $I_{k+1}^{q'}$, l'ensemble des points que la variable continue du système hybride peut valoir lors de la transition vers q' .

Le calcul de l'ensemble des successeurs continus n'est pas fondamentalement différent du calcul de l'ensemble atteignable d'un système dynamique linéaire soumis à des perturbations.

Dans la section suivante, nous montrons comment modifier l'algorithme 6, présenté dans le chapitre précédent, pour calculer l'ensemble des successeurs continus.

7.2 Calcul des successeurs continus

Pour tout état discret q , pour un ensemble de valeurs initiales $I \subseteq D_q$, l'ensemble des successeurs continus $\mathcal{X}_s^q(I)$ est l'ensemble des points $y \in D_q$, tels qu'il existe une solution de l'inclusion différentielle :

$$x'(t) = A_q x(t) + b_q + u(t), \quad x(t) \in \mathbb{R}^n, \quad \|u(t)\| \leq \mu_q, \quad (7.2.2)$$

vérifiant $x(0) \in I$ et il existe $t' \in [0, s]$, tel que

$$\forall t \in [0, t'], x(t) \in D_q \text{ et } x(t') = y.$$

D'autre part, il est intéressant de savoir si toutes les solutions de l'inclusion différentielle (7.2.2), issues de l'ensemble I , sortent du domaine D_q avant l'instant s . Nous définissons donc, la notion de temps de séjour, que l'on notera $\mathcal{T}_s^q(I)$. S'il existe une solution de l'inclusion différentielle, issue de l'ensemble I et sortant pas du domaine D_q avant l'instant s , alors $\mathcal{T}_s^q(I) = s$.

Sinon, toute solution, issue de l'ensemble I , sort du domaine D_q à un instant strictement inférieur à s . $\mathcal{T}_s^q(I)$ est la borne supérieure de ces instants de sortie. Donc, si $\mathcal{T}_s^q(I) \neq s$ alors pour tout $s' \geq s$, $\mathcal{X}_{s'}^q(I) = \mathcal{X}_s^q(I)$.

Dans la suite de cette partie, nous omettrons, par soucis de lisibilité, de noter l'indice q correspondant à l'état discret du système hybride.

L'algorithme 7 est une adaptation de l'algorithme 6 permettant le calcul d'une approximation conservatrice de l'ensemble des successeurs continus et d'une sur-approximation du temps de séjour. La principale différence avec le calcul de $\mathcal{R}_{[0,s]}^\mu(I)$, l'ensemble atteignable du système dynamique linéaire, est qu'il ne faut considérer que les trajectoires ne sortant pas du domaine D .

Q_{k+1} est une approximation conservatrice de l'ensemble des points atteignables sur l'intervalle $[kr, (k+1)r]$ par les solutions de (7.2.2) restant dans le domaine D sur $[0, (k+1)r]$. A chaque itération, on calcule l'image de Q_k par $\hat{\Phi}^\mu$, l'approximation conservatrice du flot associé à l'inclusion différentielle (7.2.2). L'intersection de cet ensemble avec le domaine D donne Q_{k+1} .

L'ensemble R_{k+1} est une approximation conservatrice de l'ensemble des points atteignables par les solutions de (7.2.2) restant dans le domaine D sur l'intervalle $[0, (k+1)r]$:

$$R_{k+1} = \bigcup_{i=1}^{i=k+1} Q_i.$$

Si $Q_{k+1} = \emptyset$, cela signifie qu'à l'instant kr , toutes les solutions sont sorties du domaine D . Par conséquent, $\mathcal{T}_s(I) \leq kr$.

Données : Un polytope I .

Résultat : Une approximation conservatrice $\hat{\mathcal{X}}_s(I)$ de l'ensemble $\mathcal{X}_s(I)$.
Une sur-approximation $\hat{T}_s(I)$ du temps de séjour $\mathcal{T}_s(I)$.

$$\varepsilon := \frac{\mu}{\|A\|} (e^{\|A\|r} - 1);$$

– *Initialisation*

$$P_1 := \hat{\mathcal{R}}_{[0,r]}(I);$$

$$Q_1 := (P_1 \oplus \mathcal{B}(0, \varepsilon)) \cap D;$$

$$R_1 := Q_1;$$

– *Boucle principale*

pour $k = 1 \dots N - 1$ **faire**

$$P_{k+1} := \Phi(Q_k, r);$$

$$Q_{k+1} := (P_{k+1} \oplus \mathcal{B}(0, \varepsilon)) \cap D;$$

$$R_{k+1} := R_k \cup Q_{k+1};$$

si $Q_{k+1} = \emptyset$ **alors**

$$\quad \quad \quad \mathbf{retourner} [\hat{\mathcal{X}}_s(I), \hat{T}_s(I)] := [R_k, kr];$$

fin

fin

retourner $[\hat{\mathcal{X}}_s(I), \hat{T}_s(I)] := [R_N, t];$

Algorithme 7: Calcul de l'ensemble des successeurs continus et du temps de séjour.

Lemme 7.2.1 Soient $\hat{\mathcal{X}}_s(I)$ et $\hat{T}_s(I)$ calculés par l'algorithme 7,

$$\mathcal{X}_s(I) \subseteq \hat{\mathcal{X}}_s(I) \text{ et } \mathcal{T}_s(I) \leq \hat{T}_s(I).$$

Preuve : Soit $y \in D$, tel qu'il existe $x(t)$, solution de (7.2.2) et $t' \in [0, r]$ tels que

$$\forall t \in [0, t'], x(t) \in D \text{ et } x(t') = y.$$

y est donc un élément de l'ensemble $\mathcal{R}_{[0,r]}^\mu(I)$ atteignable par le système dynamique linéaire sur l'intervalle $[0, r]$. Donc, d'après la proposition 6.3.3, et comme $y \in D$,

$$y \in \hat{\mathcal{R}}_{[0,r]}^\mu(I) \cap D = Q_1.$$

Supposons, que l'ensemble des points, atteignables sur l'intervalle $[kr, (k+1)r]$, par des solutions de (7.2.2) restant dans le domaine D sur $[0, (k+1)r]$, est inclus dans Q_{k+1} . L'hypothèse de récurrence est vraie pour $k = 0$.

Soit $y \in D$, tel qu'il existe $x(t)$ solution de (7.2.2) et $t' \in [kr, (k+1)r]$ tels que

$$\forall t \in [0, t'], x(t) \in D \text{ et } x(t') = y.$$

$y \in \Phi^\mu(x(t'-r), r)$, et par hypothèse $x(t'-r) \in Q_k$. D'après la proposition 6.3.2, et comme $y \in D$,

$$y \in \hat{\Phi}^\mu(Q_k, r) \cap D = Q_{k+1}.$$

Donc

$$\mathcal{X}_s(I) \subseteq \bigcup_{k=1}^{k=N} Q_k = \hat{\mathcal{X}}_s(I).$$

Supposons $Q_{k+1} = \emptyset$, il n'existe pas de solution de l'inclusion différentielle issue de l'ensemble I et restant dans D sur l'intervalle $[0, kr]$. Donc, $\mathcal{T}_s(I) \leq kr = \hat{\mathcal{T}}_s(I)$. ■

7.3 Calcul de l'ensemble atteignable d'un système hybride

Nous pouvons, maintenant, mettre en oeuvre les idées présentées au début de ce chapitre, pour le calcul de l'ensemble atteignable d'un système hybride $\mathcal{R}_{p,s}(I)$. La méthode est présentée dans l'algorithme 8, $\hat{\mathcal{X}}_s^q$ et $\hat{\mathcal{T}}_s^q$ sont calculés par l'algorithme 7.

Q_k est une partie de \mathcal{Q} contenant l'ensemble des états discrets atteignables par le système hybride en k transitions.

Pour tout état discret $q \in Q_k$, X_k^q est une approximation conservative de l'ensemble des états continus, atteignables entre la k -ième et la $(k+1)$ -ième transition, par les exécutions dont le k -ième mode est q .

P_k est une approximation conservative de l'ensemble des états atteignables entre la k -ième et la $(k+1)$ -ième transition :

$$P_k = \bigcup_{q \in Q_k} (\{q\} \times X_k^q).$$

R_{k+1} est une approximation conservative de l'ensemble des états atteignables en moins de k transitions :

$$R_{k+1} = \bigcup_{i=0}^{i=k} P_i.$$

L'ensemble calculé, $\hat{\mathcal{R}}_{p,s}(I) = R_{p+1}$, est une approximation conservative de $\mathcal{R}_{p,s}(I)$.

L'algorithme calcule également deux variables booléennes p -est-atteint et s -est-atteint, dont nous verrons l'utilité par la suite.

Théorème 7.3.1 Soit $\hat{\mathcal{R}}_{p,s}(I)$ l'ensemble calculé par l'algorithme 8,

$$\mathcal{R}_{p,s}(I) \subseteq \hat{\mathcal{R}}_{p,s}(I).$$

Preuve : Q_0 est l'ensemble des états discrets initiaux. Pour tout état discret $q \in Q_0$, l'ensemble des états continus atteignables, en un temps inférieur à s et avant la première transition est

$$\mathcal{X}_s^q(I_0^q) \subseteq \hat{\mathcal{X}}_s^q(I_0^q) = X_0^q.$$

Soit $k \in \{0, \dots, p-1\}$, supposons que pour toute exécution (τ, q, x) acceptée par le système hybride et telle que

$$\begin{cases} \tau = \{[t_i, t_{i+1}]\}_{i=0}^{i=k}, \text{ et } \forall i \in \{0, \dots, k\}, t_{i+1} - t_i \leq s \\ (q(t_0), x(t_0)) \in I, \end{cases}$$

on a

$$\exists \sigma \in Q_k, \forall t \in [t_k, t_{k+1}], q(t) = \sigma, x(t) \in X_k^\sigma.$$

Données : Un ensemble de conditions initiales $I = \bigcup_{q \in Q_0} (\{q\} \times I_0^q)$.

Résultat : Une approximation conservative $\hat{\mathcal{R}}_{p,s}(I)$ de l'ensemble atteignable $\mathcal{R}_{p,s}(I)$.

$R_0 := \emptyset;$
 p -est-atteint:=vrai;
 s -est-atteint:=faux;

pour $k = 0 \dots p$ **faire**

$P_k := \emptyset;$
 $Q_{k+1} := \emptyset;$
pour $q \in Q$ **faire**
 | $I_{k+1}^q := \emptyset;$
fin

pour $q \in Q_k$ **faire**

-Calcul des successeurs continus
 $X_k^q := \hat{\mathcal{X}}_s^q(I_k^q);$
si $\hat{\mathcal{T}}_s^q(I_k^q) = s$ **alors**
 | s -est-atteint:=vrai;
fin
 $P_k := P_k \cup (\{q\} \times X_k^q);$

-Calcul des successeurs discrets
pour $e = (q, q') \in \mathcal{E}$ **faire**
 | **si** $X_k^q \cap G_e \neq \emptyset$ **alors**
 | | $Q_{k+1} := Q_{k+1} \cup \{q'\};$
 | | $I_{k+1}^{q'} := I_{k+1}^{q'} \cup (X_k^q \cap G_e);$
 | **fin**
fin

fin

$R_{k+1} = R_k \cup P_k;$
si $R_{k+1} = R_k$ **alors**
 | p -est-atteint:=faux;
fin

fin
retourner $\hat{\mathcal{R}}_{p,s}(I) := R_{p+1}$

Algorithme 8: Calcul de l'ensemble atteignable du système hybride

L'hypothèse de récurrence est vraie pour $k = 0$.

Soit (τ, q, x) une exécution acceptée par \mathcal{H} , telle que

$$\begin{cases} \tau = \{[t_i, t_{i+1}]\}_{i=0}^{i=k+1}, \text{ et } \forall i \in \{0, \dots, k+1\}, t_{i+1} - t_i \leq s \\ (q(t_0), x(t_0)) \in I. \end{cases}$$

La variable discrète q est constante et vaut $\sigma \in \mathcal{Q}$ sur l'intervalle $[t_{k+1}, t_{k+2}]$. Soit $t \in [t_{k+1}, t_{k+2}]$, on a $q(t) = \sigma$. Par hypothèse,

$$\exists \sigma' \in \mathcal{Q}_k, q(t_{k+1}^-) = \sigma', x(t_{k+1}^-) \in X_k^{\sigma'}.$$

De plus, $x(t_{k+1}^-) \in G_e$ où $e = (\sigma', \sigma) \in \mathcal{E}$. Donc $x(t_{k+1}^-) \in X_k^{\sigma'} \cap G_e$ qui, par conséquent, n'est pas vide. On a donc $\sigma \in \mathcal{Q}_{k+1}$. D'autre part,

$$x(t_{k+1}^+) = x(t_{k+1}^-) \in X_k^{\sigma'} \cap G_e = I_{k+1}^{\sigma}.$$

Puisque $t \in [t_{k+1}, t_{k+2}]$ et $t_{k+2} - t_{k+1} \leq s$, on a

$$x(t) \in \mathcal{X}_s^{\sigma}(I_{k+1}^{\sigma}) \subseteq \hat{\mathcal{X}}_s^{\sigma}(I_{k+1}^{\sigma}) = X_{k+1}^{\sigma}.$$

Soit $(\sigma, y) \in \mathcal{R}_{p,s}(I)$, par définition, il existe une exécution (τ, q, x) acceptée par \mathcal{H} et vérifiant

$$\begin{cases} \tau = \{[t_i, t_{i+1}]\}_{i=0}^{i=N}, \text{ avec } N \leq p, \text{ et } \forall i \in \{0, \dots, k+1\}, t_{i+1} - t_i \leq s \\ (q(t_0), x(t_0)) \in I \text{ et } \exists t \in [t_N, t_{N+1}], q(t) = \sigma, x(t) = y. \end{cases}$$

D'après ce qui précède, $\sigma \in \mathcal{Q}_N$ et $y \in X_N^{\sigma}$. Donc,

$$(\sigma, y) \in P_N \subseteq \hat{\mathcal{R}}_{p,s}(I).$$

■

L'algorithme 8 permet également le calcul de deux variables booléennes, p -est-atteint et s -est-atteint. Elles permettent de décider si l'ensemble $\hat{\mathcal{R}}_{p,s}(I)$, calculé par l'algorithme 8, est une approximation conservative de l'ensemble atteignable $\mathcal{R}(I)$.

Proposition 7.3.1 *Soient p -est-atteint et s -est-atteint, les variables booléennes calculées par l'algorithme 8 :*

1. Si p -est-atteint=faux,

$$\bigcup_{p' \in \mathbb{N}} \mathcal{R}_{p',s}(I) \subseteq \hat{\mathcal{R}}_{p,s}(I).$$

2. Si s -est-atteint=faux,

$$\bigcup_{s' \in \mathbb{R}^+} \mathcal{R}_{p,s'}(I) \subseteq \hat{\mathcal{R}}_{p,s}(I).$$

3. Si p -est-atteint=faux et s -est-atteint=faux,

$$\mathcal{R}(I) \subseteq \hat{\mathcal{R}}_{p,s}(I).$$

Preuve : Remarquons d'abord, que l'on a

$$\forall p' \leq p, s' \leq s, \mathcal{R}_{p',s'}(I) \subseteq \mathcal{R}_{p,s}(I) \subseteq \hat{\mathcal{R}}_{p,s}(I).$$

- Si p -est-atteint=faux :

$$\exists k \in \{0, \dots, p\}, R_{k+1} = R_k.$$

Supposons, qu'il existe p' tel que

$$\mathcal{R}_{p',s}(I) \subseteq R_k.$$

On peut montrer (voir preuve du théorème précédent) que R_{k+1} est une approximation conservatrice de l'ensemble $\mathcal{R}_{k,s}(I)$. Donc pour tout $p' \in \{0, \dots, k\}$, $\mathcal{R}_{p',s}(I) \subseteq R_{k+1}$. Comme $R_{k+1} = R_k$, l'hypothèse de récurrence est vraie pour $p' \in \{0, \dots, k\}$.

Soit $(\sigma, y) \in \mathcal{R}_{p'+1,s}(I)$, on suppose que (σ, y) est atteignable en $p' + 1$ transitions exactement (sinon $(\sigma, y) \in \mathcal{R}_{p',s}(I) \subseteq R_k$). Il existe donc une exécution (τ, q, x) telle que

$$\begin{cases} \tau = \{[t_i, t_{i+1}]\}_{i=0}^{i=p'+1}, \text{ et } \forall i \in \{0, \dots, p' + 1\}, t_{i+1} - t_i \leq s \\ (q(t_0), x(t_0)) \in I, \exists t \in [t_{p'+1}, t_{p'+2}], q(t) = \sigma, x(t) = y. \end{cases}$$

$(q(t_{p'+1}^-), x(t_{p'+1}^-)) \in \mathcal{R}_{p',s}(I)$, donc, par hypothèse,

$$(q(t_{p'+1}^-), x(t_{p'+1}^-)) \in R_k = \bigcup_{i=0}^{i=k-1} P_i.$$

Il existe donc $i \in \{0, \dots, k-1\}$, tel que

$$(q(t_{p'+1}^-), x(t_{p'+1}^-)) \in P_i = \bigcup_{\sigma' \in Q_i} (\{\sigma\} \times X_i^{\sigma'}).$$

Donc,

$$\exists i \in \{0, \dots, k-1\}, \sigma' \in Q_i, q(t_{p'+1}^-) = \sigma', x(t_{p'+1}^-) \in X_i^{\sigma'}.$$

De plus, $x(t_{p'+1}^-) \in G_e$, où $e = (\sigma', \sigma) \in \mathcal{E}$. Donc $x(t_{p'+1}^-) \in G_e \cap X_i^{\sigma'}$, qui par conséquent n'est pas vide. On a donc $\sigma \in Q_{i+1}$, et $x(t_{p'+1}^+) \in I_{i+1}^\sigma$.

Puisque $t \in [t_{p'+1}, t_{p'+2}]$, et $t_{p'+2} - t_{p'+1} \leq s$, on a

$$y = x(t) \in \mathcal{X}_s^\sigma(I_{i+1}^\sigma) \subseteq \hat{\mathcal{X}}_s^\sigma(I_{i+1}^\sigma) = X_{i+1}^\sigma.$$

Donc,

$$(\sigma, y) \in P_{i+1} \subseteq R_{k+1} = R_k.$$

Nous pouvons conclure, en remarquant que $R_k \subseteq \hat{\mathcal{R}}_{p,s}(I)$.

- Si s -est-atteint=faux :

$$\forall k \in \{0, \dots, p\}, \forall q \in Q_k, \mathcal{T}_s^q(I_k^q) \leq \hat{\mathcal{T}}_s^q(I_k^q) < s.$$

Donc,

$$\forall k \in \{0, \dots, p\}, \forall q \in Q_k, \forall s' \geq s, \mathcal{X}_{s'}^q(I_k^q) = \mathcal{X}_s^q(I_k^q) \subseteq X_k^q.$$

En reprenant la démonstration du théorème précédent, il est facile de voir que

$$\forall s' \geq s, \mathcal{R}_{p,s'}(I) \subseteq \hat{\mathcal{R}}_{p,s}(I).$$

- Si t -est-atteint=faux et p -est-atteint=faux :

D'après ce qui précède, il est clair que

$$\forall p' \in \mathbb{N}, \forall s' \geq 0, \mathcal{R}_{p',s'}(I) \subseteq \hat{\mathcal{R}}_{p,s}(I).$$

Donc

$$\mathcal{R}(I) = \bigcup_{p' \in \mathbb{N}, s' \in \mathbb{R}^+} \mathcal{R}_{p',s'}(I) \subseteq \hat{\mathcal{R}}_{p,s}(I).$$

■

Remarque 7.3.1 *On peut modifier légèrement l'algorithme 8. En effet, si à la fin d'une itération, on a $R_{k+1} = R_k$, il est inutile de continuer. En effet, R_k est alors une approximation conservative de $\mathcal{R}_{p,s}(I)$.*

Remarque 7.3.2 *Dans l'algorithme 7, pour le calcul d'une approximation conservative de l'ensemble des successeurs continus d'un ensemble I , on a supposé que I était un polytope. Or, en général, les ensembles I_k^q de l'algorithme 8 ne sont pas des polytopes mais des unions de polytopes.*

Le calcul de $\mathcal{X}_s^q(I_k^q)$ nécessite donc un traitement préliminaire. On peut, soit décomposer I_k^q en un certain nombre de polytopes, calculer les ensembles des successeurs continus pour chaque polytope, et réunir ces ensembles; soit approcher l'ensemble I_k^q par son enveloppe convexe (voir [24]).

7.4 Exemple

Nous nous proposons d'utiliser l'algorithme 8 pour analyser la robustesse de la stratégie de contrôle du système de deux réservoirs que l'on a déjà étudié dans le chapitre 5. On a vu que ce système avait un comportement asymptotique périodique, nous avons également prouvé sa stabilité. Pour poursuivre notre étude, on veut savoir si ce comportement subsiste si l'on suppose que les dynamiques continues sont incertaines : on ne connaît pas exactement les caractéristiques des valves, l'apport d'eau dans le premier réservoir peut subir des fluctuations...

On suppose donc que les dynamiques continues du système sont données par des inclusions différentielles :

$$\forall q \in \mathcal{Q}, x'(t) = A_q x(t) + b_q + u(t), \|u(t)\| \leq \mu.$$

On veut vérifier que pour une condition initiale proche du cycle hybride, le niveau d'eau dans les réservoirs reste dans des limites acceptables. L'ensemble de conditions initiales est :

$$I = \{(q, x), q = 3, x = (x_1, x_2) \in D_3, 1.5 \leq x_1 \leq 2.5, x_2 = 1\}.$$

Nous avons calculé l'ensemble atteignable du système hybride pour les valeurs $\mu = 0$ et $\mu = 0.1$. Dans les deux cas, l'algorithme s'arrête en quatre itérations, nous donnant une approximation conservative de l'ensemble atteignable $\mathcal{R}(I)$.

Sur la figure 7.1, nous avons représenté les sous-ensembles des domaines D_3 , D_1 et D_2 atteignables par le système hybride ainsi que l'ensemble des états continus atteignables pour $\mu = 0$. Pour des conditions initiales dans I , les niveaux d'eau dans chaque réservoir restent, pour toujours, dans les limites données par cet ensemble. On vérifie donc bien que la stratégie de contrôle permet de garder le niveau d'eau dans des limites acceptables.

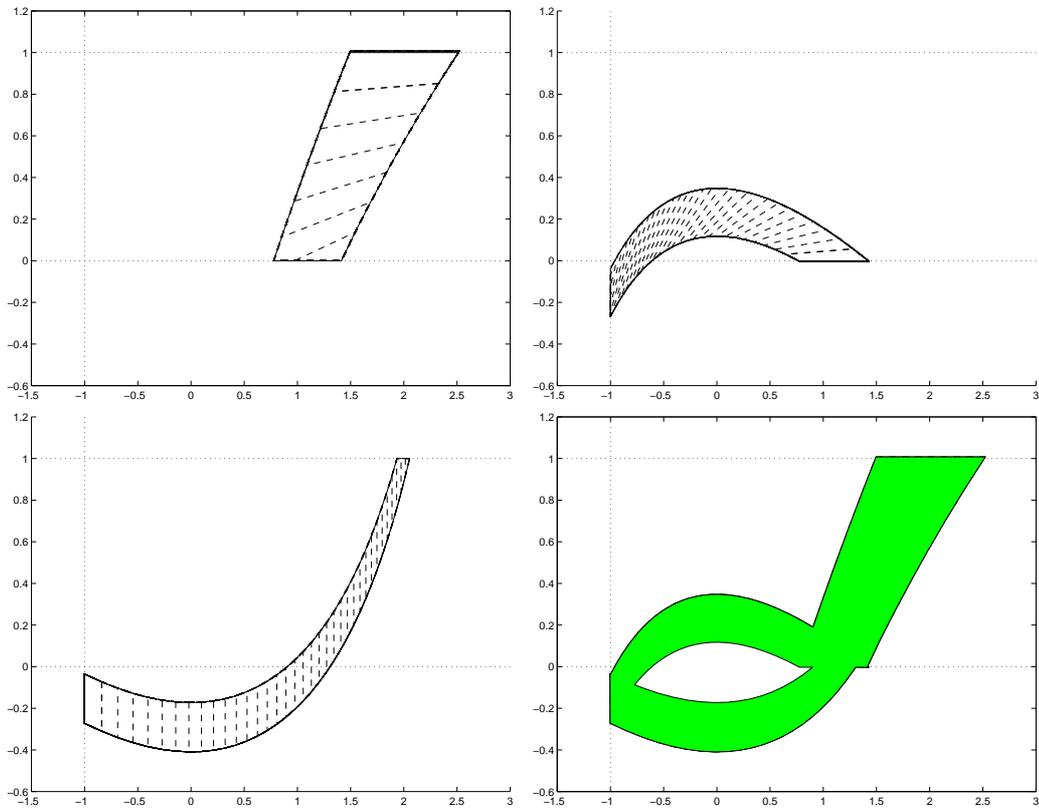


FIG. 7.1: Ensemble atteignable dans chaque mode pour le système autonome. Ensemble des états continus atteignables.

Sur la figure 7.2, nous avons représenté les sous-ensembles des domaines D_3 , D_1 et D_2 atteignables par le système hybride ainsi que l'ensemble des états continus atteignables pour $\mu = 0.1$. Pour des conditions initiales dans I , les niveaux d'eau dans chaque réservoir restent, pour toujours, dans les limites données par cet ensemble. On voit donc que la stratégie de contrôle est robuste, puisque, même si les dynamiques continues sont perturbées, les niveaux d'eau restent dans des limites acceptables. Le niveau varie entre -1 et 2.5 dans le premier réservoir et entre -0.6 et 1 dans le deuxième réservoir.

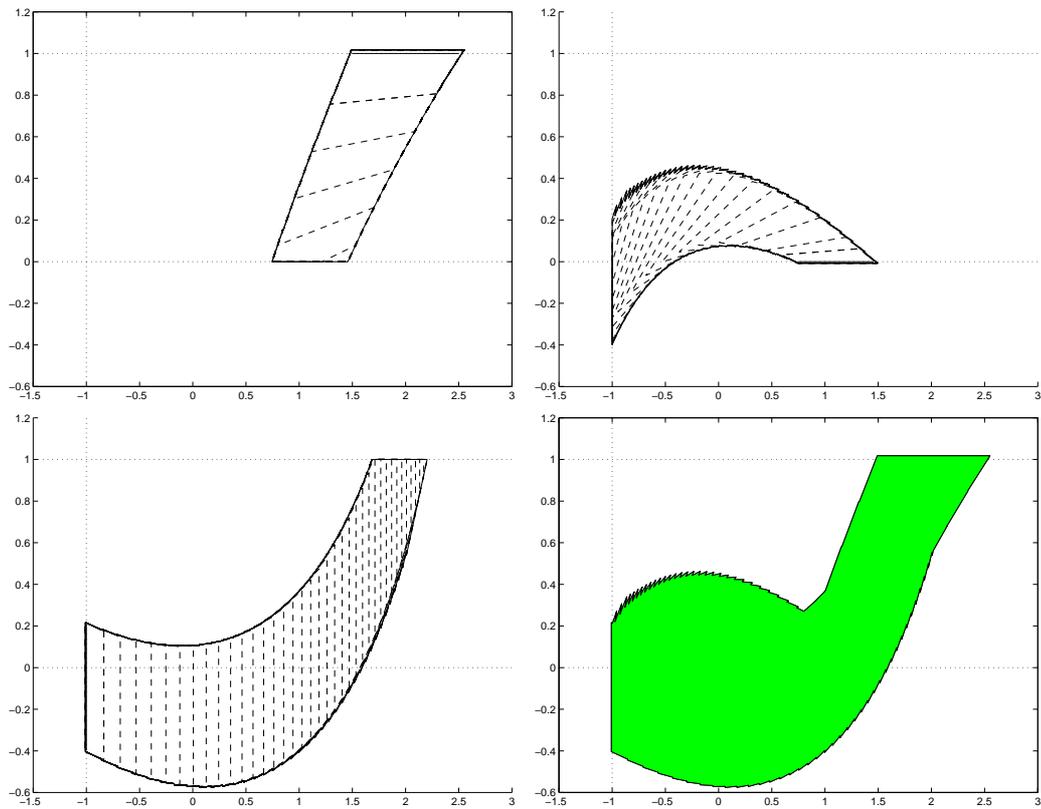


FIG. 7.2: Ensemble atteignable dans chaque mode pour le système soumis à des perturbations. Ensemble des états continus atteignables.

Sur la figure 7.3, nous avons représenté les ensembles des états continus atteignables pour $\mu = 0$ et $\mu = 0.1$.

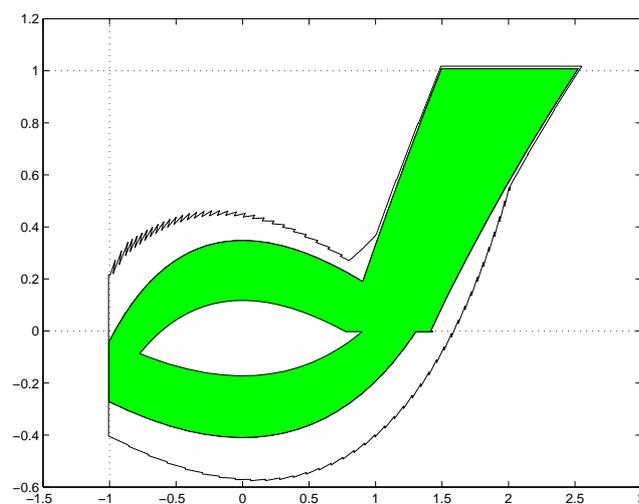


FIG. 7.3: Comparaisons des états continus atteignables pour le système autonome ou soumis à des perturbations

Quatrième partie

Contrôle des Systèmes Hybrides

Chapitre 8

Analyse Multirésolution et Théorie du Contrôle

Dans les parties précédentes, nous avons supposé que nous ne pouvions pas contrôler la dynamique du système. L'évolution continue était donnée soit par une équation différentielle autonome, soit par une équation différentielle avec un terme de perturbation sur lequel on ne peut pas agir.

Dans cette partie, on considère des systèmes hybrides où les dynamiques continues sont définies par des équations différentielles linéaires avec une entrée représentant un terme de contrôle :

$$\forall q \in \mathcal{Q}, x'(t) = A_q x(t) + B_q u(t).$$

Pour de telles équations différentielles, les fonctions u et x sont appelées respectivement entrée et sortie du système.

On s'intéressera au contrôle optimal de cette classe de systèmes hybrides et plus particulièrement au problème de planification de trajectoire : soit (t_0, q_0, x_0) une condition initiale pour le système hybride et (t_f, q_f, x_f) un état du système que l'on souhaite atteindre. Le problème de contrôle consiste à trouver une fonction u telle que, pour cette entrée, le système hybride accepte une exécution (τ, q, x) vérifiant :

$$(q(t_0), x(t_0)) = (q_0, x_0) \text{ et } (q(t_f), x(t_f)) = (q_f, x_f).$$

Généralement, une telle fonction u n'est pas unique; on cherchera donc, en plus, à minimiser un critère d'énergie :

$$\int_{t_0}^{t_f} u^T(t)u(t) dt.$$

Le contrôle optimal des systèmes hybrides linéaires par morceaux a été largement étudié. Pour des systèmes hybrides en temps discret, un cadre théorique ainsi que des outils algorithmiques, basés majoritairement sur l'optimisation combinatoire, ont été développés [13].

Pour des systèmes hybrides en temps continu (comme ceux que nous considérons), le problème devient plus complexe. Souvent, on cherche à trouver des conditions nécessaires à l'optimalité d'une entrée. Cette approche résulte en une collection de principes de

maximum hybrides [43, 84, 90]. La principale difficulté se trouve dans le grand nombre de paramètres d'optimisation : nombre de transitions [34], instants des transitions, séquence des états discrets [15, 84], et, finalement, l'entrée du système [31]. Notre étude portera essentiellement sur ce dernier point.

On supposera donc que la trajectoire temporisée $\tau = \{[t_i, t_{i+1}]\}_{i=0}^{i=N}$ (avec $t_{N+1} = t_f$), ainsi que la séquence d'états discrets $q_0 = \sigma_0, \dots, \sigma_N = q_f$ optimales sont connues. On recherche l'entrée optimale dans l'espace vectoriel de fonctions $L^2(t_0, t_f)^p$. Le problème de contrôle optimal devient alors :

$$\begin{aligned} & \text{Minimiser } \int_{t_0}^{t_f} u^T(t)u(t) dt \\ \text{sous } & \begin{cases} x(t_0) = x_0, x(t_f) = x_f, \\ x(t_i) \in G_{\sigma_i, \sigma_{i+1}}, & i \in \{1, \dots, N\}, \\ \forall t \in [t_i, t_{i+1}], x(t) \in D_{\sigma_i}, & i \in \{0, \dots, N\}, \end{cases} \end{aligned} \quad (8.0.1)$$

$$\text{avec } \forall t \in [t_i, t_{i+1}], x'(t) = A_{\sigma_i}x(t) + B_{\sigma_i}u(t), i \in \{0, \dots, N\}.$$

La principale difficulté de ce problème d'optimisation réside dans la contrainte $x(t) \in D_{\sigma_i}$. En effet, cette contrainte doit être respectée pour tout $t \in [t_i, t_{i+1}]$. Il faut donc plutôt la percevoir comme un ensemble infini de contraintes.

Une idée pour approcher l'entrée optimale est de prendre une discrétisation du temps et de résoudre le problème de contrôle optimal où les contraintes $x(t) \in D_{\sigma_i}$ ne doivent être respectées qu'aux instants de discrétisation.

Récemment, la relation entre les splines et la théorie du contrôle a été analysée [100]. Il en résulte une nouvelle classe de fonctions splines : les splines de la théorie du contrôle (*control theoretic splines*). Cette classe de fonctions a des bonnes propriétés pour résoudre une grande variété de problèmes de contrôle optimal [35, 36, 91, 100]. En particulier, la sortie optimale de la version discrétisée du problème (8.0.1) est composée de plusieurs splines de la théorie du contrôle.

La trajectoire optimale du problème (8.0.1) peut donc être vue comme la limite d'une suite de splines de la théorie du contrôle, définie par une discrétisation de plus en plus fine de l'intervalle de temps.

Un cadre théorique adapté à cette approche est celui de l'analyse multirésolution [26, 72]. Intuitivement, l'idée est la suivante : on construit une suite convergente $(f_j)_{j \in \mathbb{N}}$ d'approximations d'une fonction f . A l'échelle j , l'approximation f_j est déterminée par un échantillonnage fournissant des moyennes locales de la fonction f sur des intervalles de longueur 2^{-j} .

D'une manière plus formelle, une analyse multirésolution consiste en une séquence $(V_j)_{j \in \mathbb{N}}$ d'espaces vectoriels emboîtés. L'espace V_j regroupe toutes les approximations possibles à l'échelle j . L'approximation de f à cette résolution est définie comme la projection orthogonale de f sur V_j .

Dans ce chapitre, nous construisons une analyse multirésolution $(V_j)_{j \in \mathbb{N}}$ pour un sous-espace de $L^2(0, 1)^p$ (voir aussi [49]), permettant de construire une suite d'approximations

de l'entrée d'un système dynamique linéaire à partir d'un échantillonnage de la sortie associée. L'ensemble des splines de la théorie du contrôle, associées à une discrétisation de pas 2^{-j} de l'intervalle $[0, 1]$, est généré par les entrées de l'espace V_j .

Dans le chapitre suivant, nous verrons comment cette analyse multirésolution peut être utilisée pour le calcul d'une solution approchée du problème de contrôle optimal (8.0.1).

Mais d'abord, nous avons besoin d'introduire certaines notions de la théorie du contrôle des systèmes dynamiques linéaires.

8.1 Quelques résultats de la théorie du contrôle

On considère l'équation différentielle linéaire avec entrée :

$$x'(t) = Ax(t) + Bu(t), \quad x(0) = x_0, \quad u \in L^2(0, 1)^p \quad (8.1.2)$$

où $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^p$, A et B sont des matrices de dimensions compatibles. On a de manière équivalente,

$$\forall t \in [0, 1], \quad x(t) = e^{At}x_0 + \int_0^t e^{A(t-s)}Bu(s) ds.$$

On supposera, dans un premier temps, que $x_0 = 0$.

8.1.1 Ensemble des entrées optimales : définition, caractérisation

Notons $\mathcal{T}_{(A,B)}$ l'espace vectoriel des sorties de l'équation différentielle (8.1.2)

$$\mathcal{T}_{(A,B)} = \left\{ x : [0, 1] \rightarrow \mathbb{R}^n, \quad x(t) = \int_0^t e^{A(t-s)}Bu(s)ds, \quad u \in L^2(0, 1)^p \right\}$$

et $\mathcal{F}_{(A,B)}$ l'application linéaire qui à une entrée associe la sortie correspondante

$$\begin{aligned} \mathcal{F}_{(A,B)} : L^2(0, 1)^p &\rightarrow \mathcal{T}_{(A,B)} \\ u &\mapsto x, \quad x(t) = \int_0^t e^{A(t-s)}Bu(s)ds. \end{aligned}$$

L'application $\mathcal{F}_{(A,B)}$ n'est généralement pas bijective. Déterminer l'entrée à partir de la sortie du système est donc un problème mal posé. Pour assurer l'unicité de l'image réciproque, on doit restreindre l'ensemble des entrées.

On définit donc l'ensemble des entrées optimales, noté $\mathcal{O}_{(A,B)}$. Une entrée u est un élément de $\mathcal{O}_{(A,B)}$ si et seulement si pour toute entrée $v \in L^2(0, 1)^p$ telle que $\mathcal{F}_{(A,B)}(v) = \mathcal{F}_{(A,B)}(u)$,

$$\int_0^1 v(s)^T v(s) ds \geq \int_0^1 u(s)^T u(s) ds.$$

L'ensemble $\mathcal{O}_{(A,B)}$ permet donc de générer, à un coût minimum, la totalité des trajectoires de l'équation différentielle (8.1.2). Ainsi, la solution de n'importe quel problème de contrôle peut être recherchée dans cet ensemble.

Dans la suite de cette section, nous cherchons à donner une caractérisation de l'ensemble $\mathcal{O}_{(A,B)}$.

Lemme 8.1.1 $\mathcal{O}_{(A,B)} = \ker(\mathcal{F}_{(A,B)})^\perp$.

Preuve : Soit $u \in \mathcal{O}_{(A,B)}$, on peut l'écrire sous la forme

$$u = v + w, \text{ où } v \in \ker(\mathcal{F}_{(A,B)})^\perp, w \in \ker(\mathcal{F}_{(A,B)}).$$

On a $\mathcal{F}_{(A,B)}(v) = \mathcal{F}_{(A,B)}(u)$, et comme u est un élément de $\mathcal{O}_{(A,B)}$:

$$\begin{aligned} \int_0^1 v(s)^T v(s) ds &\geq \int_0^1 u(s)^T u(s) ds \\ &\geq \int_0^1 v(s)^T v(s) ds + \int_0^1 w(s)^T w(s) ds. \end{aligned}$$

Ainsi, $w = 0$, et $u \in \ker(\mathcal{F}_{(A,B)})^\perp$.

Soient $u \in \ker(\mathcal{F}_{(A,B)})^\perp$, $v \in L^2(0,1)^p$, telles que $\mathcal{F}_{(A,B)}(v) = \mathcal{F}_{(A,B)}(u)$. $u - v$ est dans $\ker(\mathcal{F}_{(A,B)})$, on a donc :

$$\begin{aligned} \int_0^1 v(s)^T v(s) ds &= \int_0^1 u(s)^T u(s) ds + \int_0^1 (v(s) - u(s))^T (v(s) - u(s)) ds \\ &\geq \int_0^1 u(s)^T u(s) ds. \end{aligned}$$

Donc u est dans $\mathcal{O}_{(A,B)}$. ■

Nous donnons maintenant une caractérisation de l'ensemble $\ker(\mathcal{F}_{(A,B)})$.

Lemme 8.1.2 $\ker(\mathcal{F}_{(A,B)}) = \{u \in L^2(0,1)^p \mid Bu = 0 \text{ dans } L^2(0,1)^n\}$.

Preuve : Soit $u \in \ker(\mathcal{F}_{(A,B)})$.

$$\begin{aligned} \forall t \in [0,1], \int_0^t e^{A(t-s)} Bu(s) ds = 0 &\iff \forall t \in [0,1], \int_0^t e^{-As} Bu(s) ds = 0 \\ &\iff Bu(s) = 0 \text{ presque partout sur } [0,1]. \end{aligned}$$

De manière équivalente, $Bu = 0$ dans l'espace de fonctions $L^2(0,1)^n$. ■

Avant de conclure sur la structure de l'ensemble $\mathcal{O}_{(A,B)}$, nous devons démontrer un résultat d'algèbre linéaire qui nous resservira par la suite.

Lemme 8.1.3 *Il existe η , tel que,*

$$\forall y \in \text{Im}(B^T), \exists x \in \mathbb{R}^n, \begin{cases} y &= B^T x \\ \eta \|y\| &\geq \|x\|. \end{cases}$$

où $\|\cdot\|$ dénote la norme 2.

Preuve : Notons p' la dimension de $\text{Im}(B^T)$, il existe une base orthonormale de cet ensemble composée de p' vecteurs de \mathbb{R}^p , $\{e_1 \dots e_{p'}\}$. Il existe également p' vecteurs de \mathbb{R}^n , $\{f_1 \dots f_{p'}\}$, tel que $e_i = B^T f_i$, pour $i \in \{1 \dots p'\}$.

Soit $y \in \text{Im}(B^T)$, $y = \sum_{i=1}^{i=p'} y_i e_i$. Soit $x = \sum_{i=1}^{i=p'} y_i f_i$, on a $y = B^T x$. De plus,

$$\|x\| \leq \sum_{i=1}^{i=p'} |y_i| \|f_i\| \leq \max_{i=1}^{i=p'} \|f_i\| \sum_{i=1}^{i=p'} |y_i|.$$

Remarquons que,

$$\left(\sum_{i=1}^{i=p'} |y_i| \right)^2 = \sum_{i=1}^{i=p'} \sum_{j=1}^{j=p'} |y_i| |y_j| \leq \sum_{i=1}^{i=p'} \sum_{j=1}^{j=p'} \frac{y_i^2 + y_j^2}{2} = p' \|y\|^2.$$

D'où, $\|x\| \leq \sqrt{p'} \max_{i=1}^{i=p'} \|f_i\| \|y\|$.

■

Nous pouvons, à présent, caractériser l'ensemble des entrées optimales.

Proposition 8.1.1 $\mathcal{O}_{(A,B)} = \{B^T f \mid f \in L^2(0, 1)^n\}$.

Preuve : Soit $u \in L^2(0, 1)^p$. $\ker(B)$ est le supplémentaire orthogonal de $\text{Im}(B^T)$ dans \mathbb{R}^p , il existe donc deux fonctions v et w telles que :

$$\forall t \in [0, 1], u(t) = v(t) + w(t), v(t) \in \text{Im}(B^T), w(t) \in \ker(B).$$

En particulier, pour tout $t \in [0, 1]$, $v^T(t)w(t) = 0$, donc

$$\begin{aligned} \int_0^1 u^T(s)u(s) ds &= \int_0^1 v^T(s)v(s) + w^T(s)w(s) ds \\ &= \int_0^1 v^T(s)v(s) ds + \int_0^1 w^T(s)w(s) ds. \end{aligned}$$

Donc v et w sont des éléments de $L^2(0, 1)^p$. D'après le lemme 8.1.3, il existe une fonction f telle que,

$$\forall t \in [0, 1], v(t) = B^T f(t), \text{ et } \eta \|v(t)\| \geq \|f(t)\|.$$

D'où

$$\int_0^1 f^T(s)f(s) ds \leq \eta^2 \int_0^1 v^T(s)v(s) ds.$$

f est donc un élément de $L^2(0, 1)^n$ et $v \in \{B^T f \mid f \in L^2(0, 1)^n\}$.

D'après le lemme 8.1.2, $w \in \ker(\mathcal{F}_{(A,B)})$.

Soit $f \in L^2(0, 1)^n$, on note $v = B^T f$. Soit $w \in \ker(\mathcal{F}_{(A,B)})$,

$$\int_0^1 v^T(s)w(s)ds = \int_0^1 f^T(s)Bw(s)ds = 0.$$

Donc, $\{B^T f \mid f \in L^2(0, 1)^n\}$ est le supplémentaire orthogonal, dans $L^2(0, 1)^p$, du sous-espace $\ker(\mathcal{F}_{(A,B)})$.

■

Remarque 8.1.1 $\mathcal{O}_{(A,B)} = L^2(0, 1)^p$ si et seulement si B est de rang p .

8.1.2 Contrôlabilité

Le système dynamique associé à l'équation différentielle (8.1.2) est dit contrôlable si pour tout $t \in]0, 1]$, $x_t \in \mathbb{R}^n$, il existe une entrée $u \in L^2(0, 1)^p$ menant la solution de (8.1.2) au point x_t à l'instant t :

$$\exists u \in L^2(0, 1)^p, \int_0^t e^{A(t-s)} B u(s) ds = x_t.$$

Proposition 8.1.2 [27] *Le système dynamique associé à l'équation différentielle (8.1.2) est contrôlable si et seulement si le Grammien d'atteignabilité*

$$M(t) = \int_0^t e^{A(t-s)} B B^T e^{A^T(t-s)} ds$$

est inversible pour tout $t > 0$.

Nous supposons, à partir de maintenant, que le système dynamique est contrôlable.

Proposition 8.1.3 *Pour tout $t > 0$, la matrice $M(t)$ est symétrique définie positive et strictement croissante :*

$$\forall t > 0, \tau > 0, c \in \mathbb{R}^n, c \neq 0, c^T M(t + \tau) c > c^T M(t) c.$$

Preuve : $M(t)$ est clairement symétrique. De plus, pour tout $c \in \mathbb{R}^n$,

$$c^T M(t) c = \int_0^t c^T e^{A(t-s)} B B^T e^{A^T(t-s)} c ds = \int_0^t \left\| B^T e^{A^T(t-s)} c \right\|^2 ds \geq 0.$$

D'après l'hypothèse de contrôlabilité, $M(t)$ est inversible et donc symétrique définie positive.

Soient $t > 0, \tau > 0, c \in \mathbb{R}^n, c \neq 0$,

$$\begin{aligned} c^T M(t + \tau) c &= c^T \int_0^{t+\tau} e^{A(t+\tau-s)} B B^T e^{A^T(t+\tau-s)} ds c \\ &= c^T \int_0^\tau e^{A(t+\tau-s)} B B^T e^{A^T(t+\tau-s)} ds c \\ &\quad + c^T \int_\tau^{t+\tau} e^{A(t+\tau-s)} B B^T e^{A^T(t+\tau-s)} ds c \\ &= c^T e^{At} M(\tau) e^{A^T t} c + c^T M(t) c > c^T M(t) c \end{aligned} \quad (8.1.3)$$

■

Dans la section suivante, nous développons un cadre multirésolution pour le contrôle des systèmes dynamiques linéaires.

8.2 Une analyse multirésolution de $\mathcal{O}_{(A,B)}$

Nous présentons d'abord brièvement les définitions fondamentales de la théorie de l'analyse multirésolution. Ensuite, nous verrons comment cette théorie peut s'appliquer au contrôle des systèmes linéaires.

8.2.1 Analyse multirésolution et base d'ondelettes

Nous utilisons une notion d'analyse multirésolution similaire à celle présentée dans [92]. Notre définition est un peu moins générale mais suffisante pour notre problème. Soit \mathcal{O} un sous-espace de $L^2(0, 1)^p$.

Définition 8.2.1 (Analyse Multirésolution) *Une analyse multirésolution de \mathcal{O} est une séquence de sous-espaces fermés $(V_j)_{j \in \mathbb{N}}$ vérifiant :*

1. $V_j \subseteq V_{j+1}$;
2. $\bigcup_{j \in \mathbb{N}} V_j$ est dense dans \mathcal{O} ;
3. pour tout $j \in \mathbb{N}$, V_j a une base orthogonale $\{\varphi_{j,k}, k \in \mathcal{K}(j)\}$ où $\mathcal{K}(j)$ est un ensemble d'indices.

Ainsi, une analyse multirésolution consiste en une séquence de sous-espaces approchant \mathcal{O} de plus en plus finement. Soit u un élément de \mathcal{O} , l'approximation de u à l'échelle j est la projection orthogonale de u sur V_j .

La notion d'analyse multirésolution est étroitement liée à celle d'ondelette. Les ondelettes sont définies comme des fonctions de base du supplémentaire orthogonal de V_j dans V_{j+1} .

Définition 8.2.2 (Base d'Ondelettes) *Un ensemble de fonctions de \mathcal{O} , $\{\psi_{j,m}, m \in \mathcal{M}(j)\}$ est une base d'ondelettes orthonormale si*

1. Le sous-espace de \mathcal{O} , $W_j = \text{vect}(\{\psi_{j,m}, m \in \mathcal{M}(j)\})$ est le supplémentaire orthogonal de V_j dans V_{j+1} .
2. L'ensemble de fonctions $\{\psi_{j,m}, j \in \mathbb{N}, m \in \mathcal{M}(j)\} \cup \{\varphi_{0,k}, k \in \mathcal{K}(0)\}$ est une base orthonormale de \mathcal{O} .

Dans ce cas, toute fonction $u \in \mathcal{O}$ peut s'écrire comme une combinaison linéaire infinie d'ondelettes :

$$u = \sum_{k \in \mathcal{K}(0)} \alpha_{0,k} \varphi_{0,k} + \sum_{j \in \mathbb{N}} \sum_{m \in \mathcal{M}(j)} \beta_{j,m} \psi_{j,m}$$

L'approximation de u , à l'échelle J , notée u_J est obtenue en tronquant la somme à l'indice $J - 1$:

$$u_J = \sum_{k \in \mathcal{K}(0)} \alpha_{0,k} \varphi_{0,k} + \sum_{j=0}^{J-1} \sum_{m \in \mathcal{M}(j)} \beta_{j,m} \psi_{j,m}$$

Dans le paragraphe suivant, nous nous proposons de construire une analyse multirésolution de l'ensemble des entrées optimales $\mathcal{O}_{(A,B)}$.

8.2.2 Analyse multirésolution de la théorie du contrôle

Nous voulons construire une séquence de sous-espaces d'approximation de $\mathcal{O}_{(A,B)}$, adaptée à la résolution d'une version discrétisée du problème de contrôle optimal (8.0.1). Dans ce but, il est intéressant qu'un problème de contrôle, associé à une discrétisation de l'intervalle $[0, 1]$ de pas 2^{-j} , ait sa solution dans l'ensemble d'approximation V_j .

On souhaite donc définir, l'approximation u_j , à l'échelle j , d'une fonction $u \in \mathcal{O}_{(A,B)}$ comme la solution du problème de contrôle optimal :

$$\begin{aligned} & \text{Minimiser } \int_0^1 v(s)^T v(s) ds \\ & \text{sous } x(k/2^j) = y(k/2^j), \quad k \in \{1, \dots, 2^j\}, \\ & \text{où } \begin{cases} x = \mathcal{F}_{(A,B)}(u) \\ y = \mathcal{F}_{(A,B)}(v). \end{cases} \end{aligned} \tag{8.2.4}$$

On peut montrer, [100], que u_j est l'unique élément de l'ensemble

$$V_j = \left\{ B^T e^{-A^T t} f_j(t), \quad f_j : [0, 1] \rightarrow \mathbb{R}^n, \text{ constante sur chaque } [(k-1)/2^j, k/2^j[\right\}$$

vérifiant les contraintes du problème (8.2.4).

Théorème 8.2.1 *La séquence de sous-espaces emboîtés $(V_j)_{j \in \mathbb{N}}$ est une analyse multirésolution de l'ensemble des entrées optimales $\mathcal{O}_{(A,B)}$.*

Preuve : La propriété $V_j \subseteq V_{j+1}$ est évidente. Pour tout $j \in \mathbb{N}$, V_j est un sous-espace vectoriel de $\mathcal{O}_{(A,B)}$ de dimension finie, on peut donc construire une base orthogonale de V_j .

Prouvons maintenant que $\bigcup_{j \in \mathbb{N}} V_j$ est dense dans $\mathcal{O}_{(A,B)}$. Soit u un élément de $\mathcal{O}_{(A,B)}$, d'après la proposition 8.1.1, $u = B^T f$ où f est une fonction de $L^2(0, 1)^n$. Or, l'ensemble des fonctions constantes par morceaux sur les subdivisions dyadiques de l'intervalle $[0, 1]$ est dense dans $L^2(0, 1)^n$ [72].

Soit $\varepsilon > 0$, il existe $j \in \mathbb{N}$ et f_j une fonction constante par morceaux sur la subdivision de $[0, 1]$ de pas 2^{-j} telle que

$$\int_0^1 \|f(s) - f_j(s)\|^2 ds \leq \varepsilon.$$

Soit $u_j = B^T f_j$, il est clair que u_j est un élément de V_j . De plus,

$$\begin{aligned} \int_0^1 \|u(s) - u_j(s)\|^2 ds &= \int_0^1 \|B^T (f(s) - f_j(s))\|^2 ds \\ &\leq \|B^T\|^2 \int_0^1 \|f(s) - f_j(s)\|^2 ds \leq \|B^T\|^2 \varepsilon. \end{aligned}$$

■

Pour avoir une représentation hiérarchique des fonctions de $\mathcal{O}_{(A,B)}$, il nous faut définir une base d'ondelettes pour l'analyse multirésolution $(V_j)_{j \in \mathbb{N}}$.

8.2.3 Ondelettes de la théorie du contrôle

Notons W_j le supplémentaire orthogonal de V_j dans V_{j+1} .

Lemme 8.2.1 *Un élément u de V_{j+1} est dans le sous-espace W_j si et seulement si $\mathcal{F}_{(A,B)}(u)$ s'annule aux points $\{k/2^j, k \in \{1, \dots, 2^j\}\}$.*

Preuve : Soit v un élément de V_j ,

$$v(t) = B^T e^{-A^T t} f_k, \text{ si } t \in [(k-1)/2^j, k/2^j[, k \in \{1, \dots, 2^j\}.$$

Soit $w \in V_{j+1}$, tel que $\mathcal{F}_{(A,B)}(w)$ s'annule aux points $\{k/2^j, k \in \{1, \dots, 2^j\}\}$.

$$\forall k \in \{1, \dots, 2^j\}, \int_0^{k/2^j} e^{A(k/2^j-s)} Bw(s) ds = 0.$$

Donc, pour tout $k \in \{1, \dots, 2^j\}$,

$$\begin{aligned} \int_{(k-1)/2^j}^{k/2^j} e^{-As} Bw(s) ds &= e^{-Ak/2^j} \int_0^{k/2^j} e^{A(k/2^j-s)} Bw(s) ds \\ &\quad - e^{-A(k-1)/2^j} \int_0^{(k-1)/2^j} e^{A((k-1)/2^j-s)} Bw(s) ds = 0. \end{aligned}$$

Par conséquent,

$$\int_0^1 v^T(s) w(s) ds = \sum_{k=1}^{k=2^j} f_k^T \int_{(k-1)/2^j}^{k/2^j} e^{-As} Bw(s) ds = 0.$$

On a donc

$$\{w \in V_{j+1}, \mathcal{F}_{(A,B)}(w) \text{ s'annule aux points } \{k/2^j, k \in \{1, \dots, 2^j\}\}\} \subseteq W_j.$$

De plus, il est facile de montrer que ces espaces sont de même dimension (c'est à dire $2^j n$) et donc égaux. ■

Nous pouvons maintenant construire une base d'ondelettes orthonormale pour l'analyse multirésolution $(V_j)_{j \in \mathbb{N}}$. Il existe une infinité de bases de $\mathcal{O}_{(A,B)}$ vérifiant la définition 8.2.2. On restreindra l'ensemble des choix possibles en imposant certaines propriétés supplémentaires que devra vérifier la base d'ondelettes.

V_0 est de dimension n , W_j est de dimension $2^j n$, une telle base de $\mathcal{O}_{(A,B)}$ peut être trouvée sous la forme :

- $\{\varphi^l, l \in \{1, \dots, n\}\}$ est une base orthonormale de V_0 ;
- Pour tout $j \in \mathbb{N}$, $\{\psi_{j,m}^l, m \in \{0, \dots, 2^j - 1\}, l \in \{1, \dots, n\}\}$ est une base orthonormale de W_j .

Construction d'une base de V_0

On recherche φ^l sous la forme :

$$\varphi^l(t) = B^T e^{A^T(1-t)} f^l, \quad l \in \{1, \dots, n\}$$

où $f^l \in \mathbb{R}^n$. Il est clair que $\varphi^l \in V_0$. Soient $l, l' \in \{1, \dots, n\}$, on doit avoir

$$\delta_{l,l'} = \int_0^1 \varphi^{lT}(s) \varphi^{l'}(s) ds = f^{lT} M(1) f^{l'}.$$

$M(1)$ est symétrique définie positive donc il existe une base orthonormale de \mathbb{R}^n de vecteurs propres $\{e_0^1, \dots, e_0^n\}$ associés aux valeurs propres $\lambda_0^1 \geq \dots \geq \lambda_0^n > 0$. Posons

$$f^l = \frac{1}{\sqrt{\lambda_0^l}} e_0^l, \quad l \in \{1 \dots n\}.$$

On a alors,

$$\int_0^1 \varphi^{lT}(s) \varphi^{l'}(s) ds = \frac{e_0^{lT}}{\sqrt{\lambda_0^l}} M(1) \frac{e_0^{l'}}{\sqrt{\lambda_0^{l'}}} = \sqrt{\frac{\lambda_0^{l'}}{\lambda_0^l}} e_0^{lT} e_0^{l'} = \delta_{l,l'}.$$

On peut énoncer le résultat suivant :

Lemme 8.2.2 $\{\varphi^l, l \in \{1, \dots, n\}\}$ est une base orthonormale de V_0 .

Construction d'une base de W_j

Soit $j \in \mathbb{N}$, on recherche une base de W_j vérifiant les propriétés suivantes :

– **Localisation** : pour tout $m \in \{0, \dots, 2^j - 1\}$, pour tout $l \in \{1, \dots, n\}$,

$$\forall t \in [0, m/2^j] \cup [(m+1)/2^j, 1], \quad \psi_{j,m}^l(t) = 0.$$

– **Translation** : pour tout $m \in \{0, \dots, 2^j - 1\}$, pour tout $l \in \{1, \dots, n\}$,

$$\forall t \in [m/2^j, (m+1)/2^j], \quad \psi_{j,m}^l(t) = \psi_{j,0}^l(t - m/2^j).$$

La première propriété permet d'analyser ou de synthétiser une entrée d'un point de vue local. La seconde permet d'éviter certains calculs inutiles : à chaque échelle, il suffit de calculer n fonctions pour avoir une base de W_j .

Construisons les fonctions $\{\psi_{j,0}^l, l \in \{1, \dots, n\}\}$. $\psi_{j,0}^l \in V_{j+1}$, on peut donc l'écrire sous la forme :

$$\psi_{j,0}^l(t) = \begin{cases} B^T e^{A^T(1/2^{j+1}-t)} g_j^l & \text{sur } [0, 1/2^{j+1}[\\ B^T e^{A^T(1/2^j-t)} h_j^l & \text{sur } [1/2^{j+1}, 1/2^j[\\ 0 & \text{sur } [1/2^j, 1[. \end{cases}$$

On veut aussi $\psi_{j,0}^l \in W_j$, donc $\mathcal{F}_{(A,B)}(\psi_{j,0}^l)$ s'annule en $1/2^j$:

$$\begin{aligned} \int_0^{1/2^j} e^{A(1/2^j-s)} B \psi_{j,0}^l(s) ds &= \int_0^{1/2^{j+1}} e^{A(1/2^j-s)} B B^T e^{A^T(1/2^{j+1}-t)} g_j^l ds \\ &\quad + \int_{1/2^{j+1}}^{1/2^j} e^{A(1/2^j-s)} B B^T e^{A^T(1/2^j-t)} h_j^l ds \\ &= e^{A/2^{j+1}} M(1/2^{j+1}) g_j^l + M(1/2^{j+1}) h_j^l = 0. \end{aligned}$$

On a donc

$$h_j^l = -M^{-1}(1/2^{j+1})e^{A/2^{j+1}}M(1/2^{j+1})g_j^l.$$

La condition d'orthonormalité donne pour $l, l' \in \{1, \dots, n\}$,

$$\begin{aligned} \int_0^1 \psi_{j,0}^{lT}(s)\psi_{j,0}^{l'}(s)ds &= \int_0^{1/2^{j+1}} g_j^{lT} e^{A\left(\frac{1}{2^{j+1}}-s\right)} BB^T e^{A\left(\frac{1}{2^{j+1}}-s\right)} g_j^{l'} ds \\ &\quad + \int_{1/2^{j+1}}^{1/2^j} h_j^{lT} e^{A\left(\frac{1}{2^j}-s\right)} BB^T e^{A\left(\frac{1}{2^j}-s\right)} h_j^{l'} ds \\ &= g_j^{lT} M(1/2^{j+1}) g_j^{l'} + h_j^{lT} M(1/2^{j+1}) h_j^{l'} = \delta_{l,l'}. \end{aligned}$$

En remplaçant h_j^l par son expression en fonction g_j^l , on arrive à

$$g_j^{lT} \left[M(1/2^{j+1}) + M(1/2^{j+1}) e^{\frac{A^T}{2^{j+1}}} M^{-1}(1/2^{j+1}) e^{\frac{A}{2^{j+1}}} M(1/2^{j+1}) \right] g_j^{l'} = \delta_{l,l'}.$$

On choisit

$$g_j^l = \frac{1}{\sqrt{\lambda_{j+1}^l}} M^{-1} \left(\frac{1}{2^{j+1}} \right) e_{j+1}^l,$$

où $0 < \lambda_{j+1}^1 \leq \dots \leq \lambda_{j+1}^n$ sont les valeurs propres de la matrice symétrique définie positive

$$e^{A^T/2^{j+1}} M^{-1} \left(\frac{1}{2^{j+1}} \right) e^{A/2^{j+1}} + M^{-1} \left(\frac{1}{2^{j+1}} \right),$$

et $\{e_{j+1}^1, \dots, e_{j+1}^n\}$ est une base orthonormale de \mathbb{R}^n composée de vecteurs propres associés. On a alors

$$\begin{aligned} \int_0^1 \psi_{j,0}^{lT}(s)\psi_{j,0}^{l'}(s)ds &= \frac{e_{j+1}^{lT}}{\sqrt{\lambda_{j+1}^l}} \left[M^{-1}(1/2^{j+1}) + e^{\frac{A^T}{2^{j+1}}} M^{-1}(1/2^{j+1}) e^{\frac{A}{2^{j+1}}} \right] \frac{e_{j+1}^{l'}}{\sqrt{\lambda_{j+1}^{l'}}} \\ &= \sqrt{\frac{\lambda_{j+1}^{l'}}{\lambda_{j+1}^l}} e_{j+1}^{lT} e_{j+1}^{l'} = \delta_{l,l'}. \end{aligned}$$

Lemme 8.2.3 Pour tout $j \in \mathbb{N}$, $\{\psi_{j,m}^l, m \in \{0, \dots, 2^j - 1\}, l \in \{1, \dots, n\}\}$ est une base orthonormale de W_j .

Preuve : Soient $m \in \{0, \dots, 2^j - 1\}$, $l \in \{1, \dots, n\}$, il est clair que $\psi_{j,m}^l \in V_{j+1}$.

Montrons que $\mathcal{F}_{(A,B)}(\psi_{j,m}^l)$ s'annule aux points $k/2^j$, $k \in \{1, \dots, 2^j\}$.

Pour $k \leq m$, $\psi_{j,m}^l = 0$ sur l'intervalle $[0, k/2^j[$ donc, $\mathcal{F}_{(A,B)}(\psi_{j,m}^l) = 0$ sur $[0, k/2^j]$.

Pour $k = m + 1$,

$$\begin{aligned} \int_0^{(m+1)/2^j} e^{A\left(\frac{m+1}{2^j}-s\right)} B \psi_{j,m}^l(s) ds &= \int_{m/2^j}^{(m+1)/2^j} e^{A\left(\frac{m+1}{2^j}-s\right)} B \psi_{j,0}^l(s - m/2^j) ds \\ &= \int_0^{1/2^j} e^{A(1/2^j-s)} B \psi_{j,0}^l(s) ds = 0. \end{aligned}$$

Pour $k > m + 1$, $\psi_{j,m}^l = 0$ sur l'intervalle $[(m + 1)/2^j, k/2^j[$ donc, $\mathcal{F}_{(A,B)}(\psi_{j,m}^l) = 0$ sur $[(m + 1)/2^j, k/2^j]$. Donc $\psi_{j,m}^l \in W_j$.

Montrons que les $\psi_{j,m}^l$ forment une famille orthonormale. Soient $m, m' \in \{0, \dots, 2^j - 1\}$, $l, l' \in \{1, \dots, n\}$.

Si $m \neq m'$, alors $\psi_{j,m}^l$ et $\psi_{j,m'}^{l'}$ ont des supports disjoints, donc elles sont orthogonales.

Si $m = m'$,

$$\begin{aligned} \int_0^1 \psi_{j,m}^{lT}(s) \psi_{j,m}^{l'}(s) ds &= \int_{m/2^j}^{(m+1)/2^j} \psi_{j,0}^{lT}(s - m/2^j) \psi_{j,0}^{l'}(s - m/2^j) ds \\ &= \int_0^{1/2^j} \psi_{j,0}^{lT}(s) \psi_{j,0}^{l'}(s) ds = \delta_{l,l'}. \end{aligned}$$

Donc $\{\psi_{j,m}^l, m \in \{0, \dots, 2^j - 1\}, l \in \{1, \dots, n\}\}$ est une famille orthonormale de $2^j n$ éléments du sous-espace W_j qui est de dimension $2^j n$. C'est donc une base orthonormale. ■

On peut résumer les résultats précédents dans le théorème suivant.

Théorème 8.2.2 Soient

$$\varphi^l(t) = B^T e^{A^T(1-t)} f^l, \quad l \in \{1, \dots, n\}$$

avec

$$f^l = \frac{1}{\sqrt{\lambda_0^l}} e_0^l,$$

où $\lambda_0^1 \geq \dots \geq \lambda_0^n > 0$ sont les valeurs propres de la matrice $M(1)$, et $\{e_0^1, \dots, e_0^n\}$ est la base orthonormale de \mathbb{R}^n composée des vecteurs propres associés.

$$\psi_{j,m}^l(t) = \begin{cases} 0 & \text{sur } [0, \frac{m}{2^j}[\\ B^T e^{A^T(\frac{2m+1}{2^j} - t)} g_j^l & \text{sur } [\frac{m}{2^j}, \frac{2m+1}{2^j}[\\ B^T e^{A^T(\frac{m+1}{2^j} - t)} h_j^l & \text{sur } [\frac{2m+1}{2^j}, \frac{m+1}{2^j}[\\ 0 & \text{sur } [\frac{m+1}{2^j}, 1[\end{cases}$$

$$j \in \mathbb{N}, \quad m \in \{0, \dots, 2^j - 1\}, \quad l \in \{1, \dots, n\}$$

avec

$$\begin{aligned} g_j^l &= \frac{1}{\sqrt{\lambda_{j+1}^l}} M^{-1} \left(\frac{1}{2^{j+1}} \right) e_{j+1}^l, \\ h_j^l &= \frac{-1}{\sqrt{\lambda_{j+1}^l}} M^{-1} \left(\frac{1}{2^{j+1}} \right) e^{A/2^{j+1}} e_{j+1}^l, \end{aligned}$$

où $0 < \lambda_{j+1}^1 \leq \dots \leq \lambda_{j+1}^n$ sont les valeurs propres de la matrice

$$e^{A^T/2^{j+1}} M^{-1} \left(\frac{1}{2^{j+1}} \right) e^{A/2^{j+1}} + M^{-1} \left(\frac{1}{2^{j+1}} \right),$$

et $\{e_{j+1}^1, \dots, e_{j+1}^l\}$ est une base orthonormale de \mathbb{R}^n composée des vecteurs propres associés.

L'ensemble de fonctions $\{\psi_{j,m}^l, j \in \mathbb{N}, m \in \{0, \dots, 2^j - 1\}, l \in \{1, \dots, n\}\}$ est une base d'ondelettes orthonormale de l'analyse multirésolution $\{V_j\}_{j \in \mathbb{N}}$.

Preuve : Le théorème est une conséquence immédiate du théorème 8.2.1 et des lemmes 8.2.2 et 8.2.3. ■

8.2.4 Base des fonctions sorties

La base de l'ensemble des entrées du système dynamique définit de manière implicite une base de l'ensemble des sorties du système :

$$\begin{aligned} \chi^l &= \mathcal{F}_{(A,B)}(\varphi^l) \\ \xi_{j,m}^l &= \mathcal{F}_{(A,B)}(\psi_{j,m}^l) \\ j \in \mathbb{N}, m \in \{0, \dots, 2^j - 1\}, l \in \{1, \dots, n\}. \end{aligned}$$

Sur la figure 8.1, nous avons tracé quelques fonctions de la base d'ondelettes de l'ensemble des entrées ainsi que les sorties associées pour l'équation différentielle scalaire $x' = -x + u$.

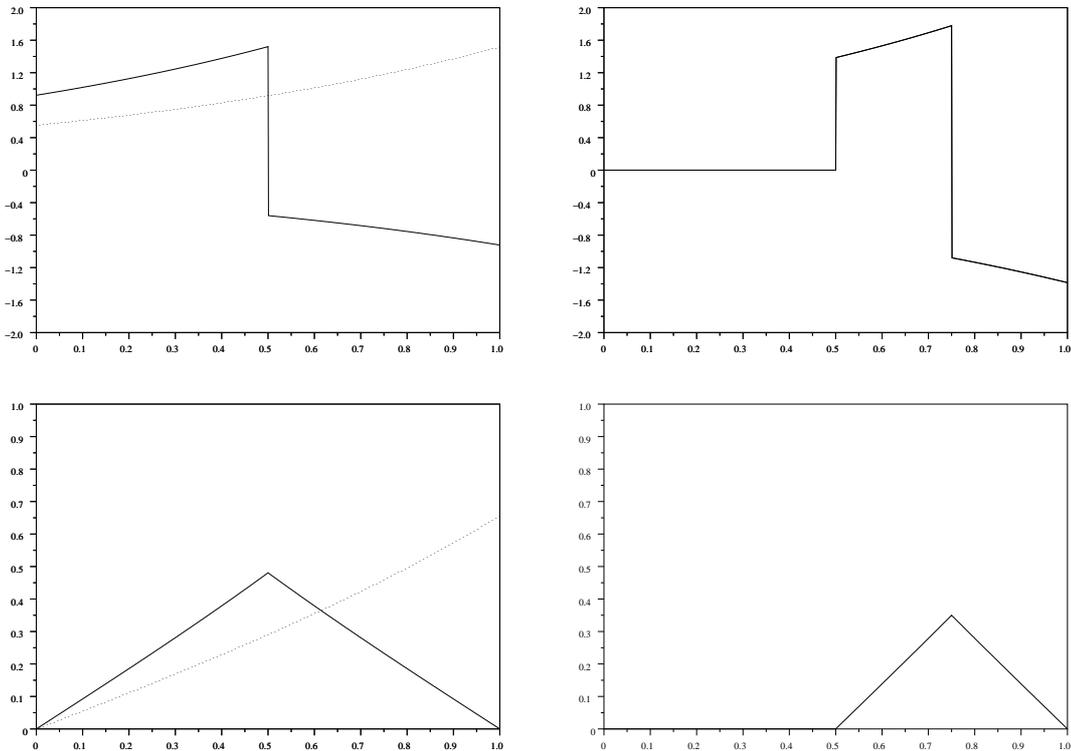


FIG. 8.1: Les fonctions φ^1 , $\psi_{0,0}^1$ (en haut à gauche), $\psi_{1,1}^1$ (en haut à droite) et les sorties associées χ^1 , $\xi_{0,0}^1$ (en bas à gauche), $\xi_{1,1}^1$ (en bas à droite) pour l'équation différentielle scalaire $x' = -x + u$.

Soit u un élément de $\mathcal{O}_{(A,B)}$, d'après le théorème 8.2.2, il existe des coefficients $\{\alpha^l, l \in \{1, \dots, n\}\}$, $\{\beta_{j,m}^l, j \in \mathbb{N}, m \in \{0, \dots, 2^j - 1\}, l \in \{1, \dots, n\}\}$ tels que :

$$u = \sum_{l=1}^{l=n} \alpha^l \varphi^l + \sum_{j=0}^{j=\infty} \sum_{m=0}^{m=2^j-1} \sum_{l=1}^{l=n} \beta_{j,m}^l \psi_{j,m}^l.$$

Proposition 8.2.1 Soit $x = \mathcal{F}_{(A,B)}(u)$, pour tout $t \in [0, 1]$,

$$x(t) = \sum_{l=1}^{l=n} \alpha^l \chi^l(t) + \sum_{j=0}^{j=\infty} \sum_{m=0}^{m=2^j-1} \sum_{l=1}^{l=n} \beta_{j,m}^l \xi_{j,m}^l(t).$$

Preuve : Soit $J \in \mathbb{N}$, soit u_J la projection orthogonale de u sur V_J :

$$u_J = \sum_{l=1}^{l=n} \alpha^l \varphi^l + \sum_{j=0}^{j=J-1} \sum_{m=0}^{m=2^j-1} \sum_{l=1}^{l=n} \beta_{j,m}^l \psi_{j,m}^l.$$

On a,

$$\lim_{J \rightarrow \infty} \int_0^1 \|u(s) - u_J(s)\|^2 ds = 0.$$

Soient $x_J = \mathcal{F}_{(A,B)}(u_J)$, $c \in \mathbb{R}^n$ tel que $\|c\| = 1$ et $t \in [0, 1]$,

$$\begin{aligned} |c^T(x(t) - x_J(t))|^2 &= \left| \int_0^t c^T e^{A(t-s)} B(u(s) - u_J(s)) ds \right|^2 \\ &\leq \int_0^t c^T e^{A(t-s)} B B^T e^{A^T(t-s)} c ds \int_0^t \|u(s) - u_J(s)\|^2 ds \\ &\leq c^T M(t) c \int_0^1 \|u(s) - u_J(s)\|^2 ds \\ &\leq c^T M(1) c \int_0^1 \|u(s) - u_J(s)\|^2 ds \end{aligned}$$

Comme l'inégalité est valable pour tout $c \in \mathbb{R}^n$, tel que $\|c\| = 1$, on a

$$\lim_{J \rightarrow \infty} \|x(t) - x_J(t)\| = 0.$$

On conclut en remarquant que

$$x_J(t) = \sum_{l=1}^{l=n_x} \alpha^l \chi^l(t) + \sum_{j=0}^{j=J-1} \sum_{m=0}^{m=2^j-1} \sum_{l=1}^{l=n} \beta_{j,m}^l \xi_{j,m}^l(t).$$

■

Les fonctions sorties associées à la base d'ondelettes de $\mathcal{O}_{(A,B)}$ forment donc également une base de l'espace vectoriel des sorties du système dynamique.

De plus, elles vérifient des propriétés intéressantes de hiérarchisation et de localisation de l'information ainsi que d'invariance par translation.

Proposition 8.2.2

– **Hierarchisation** : pour tout $j \in \mathbb{N}$, $m \in \{0, \dots, 2^j - 1\}$, $l \in \{1, \dots, n\}$,

$$\forall J \leq j, k \in \{1, \dots, 2^J\}, \xi_{j,m}^l(k/2^J) = 0.$$

– **Localisation** : pour tout $j \in \mathbb{N}$, $m \in \{0, \dots, 2^j - 1\}$, $l \in \{1, \dots, n\}$,

$$\forall t \in [0, m/2^j] \cup [(m+1)/2^j, 1], \xi_{j,m}^l(t) = 0.$$

– **Translation** : pour tout $j \in \mathbb{N}$, $m \in \{0, \dots, 2^j - 1\}$, $l \in \{1, \dots, n\}$,

$$\forall t \in [m/2^j, (m+1)/2^j], \xi_{j,m}^l(t) = \xi_{j,0}^l(t - m/2^j).$$

Preuve : Soient $j \in \mathbb{N}$, $m \in \{0, \dots, 2^j - 1\}$, $l \in \{1, \dots, n\}$.

Soient $J \leq j$, $k \in \{1, \dots, 2^J\}$, $k/2^J = k(2^j - J)/2^j$. D'après le théorème 8.2.2, $\psi_{j,m}^l$ est un élément de W_j donc $\xi_{j,m}^l = \mathcal{F}_{(A,B)}(\psi_{j,m}^l)$ s'annule aux points $\{q/2^j, q \in \{1, \dots, 2^j\}\}$ et donc en particulier aux points $\{k/2^J, k \in \{1, \dots, 2^J\}\}$. On a donc également,

$$\xi_{j,m}^l\left(\frac{m}{2^j}\right) = \xi_{j,m}^l\left(\frac{m+1}{2^j}\right) = 0.$$

Or

$$\forall t \in [0, m/2^j] \cup [(m+1)/2^j, 1], \psi_{j,m}^l(t) = 0.$$

En remarquant que $\xi_{j,m}^l$ est la sortie du système dynamique associée à l'entrée $\psi_{j,m}^l$:

$$\forall t \in [0, m/2^j] \cup [(m+1)/2^j, 1], \xi_{j,m}^l(t) = 0.$$

Soit $t \in [m/2^j, (m+1)/2^j]$,

$$\begin{aligned} \xi_{j,m}^l(t) &= \int_0^t e^{A(t-s)} B \psi_{j,m}^l(s) ds = \int_{m/2^j}^t e^{A(t-s)} B \psi_{j,m}^l(s) ds \\ &= \int_{m/2^j}^t e^{A(t-s)} B \psi_{j,0}^l(s - m/2^j) ds = \int_0^{t-m/2^j} e^{A(t-m/2^j-s)} B \psi_{j,0}^l(s) ds \\ &= \xi_{j,0}^l(t - m/2^j). \end{aligned}$$

■

Ces propriétés sont intéressantes pour traiter des problèmes de synthèse de contrôle. La première propriété signifie que la valeur de la sortie aux points d'une discrétisation de l'intervalle $[0, 1]$ de pas 2^{-J} est donnée par les coefficients associés aux échelles inférieures à J . La deuxième garantit qu'un coefficient associé à l'échelle j modifie la trajectoire du système dynamique sur un intervalle de longueur 2^{-j} . La troisième permet de ne générer que n fonctions de base à chaque échelle. Les autres sont obtenues par translation.

D'autres propriétés, ainsi qu'une application à la compression des signaux d'entrée ont été montrées dans [49].

Dans le chapitre suivant, nous montrons comment la base d'ondelettes des fonctions d'entrée et sa base induite des fonctions de sortie peuvent être utilisées pour calculer une approximation de la solution du problème de contrôle optimal hybride (8.0.1).

Chapitre 9

Contrôle Optimal des Systèmes Hybrides

On s'intéresse au problème de planification de trajectoire (8.0.1) présenté au début du chapitre précédent. On suppose que pour tout $i \in \{0, \dots, N\}$, $t_i < t_{i+1}$. On reformule (8.0.1) en pratiquant une adimensionnalisation de la variable de temps :

$$s = i + \frac{t - t_i}{t_{i+1} - t_i}, \quad i \in \{0, \dots, N\}.$$

On arrive, après quelques transformations, à un problème de contrôle optimal équivalent :

$$\begin{aligned} & \text{Minimiser } \sum_{i=0}^{i=N} \frac{1}{t_{i+1} - t_i} \int_i^{i+1} u^T(s) u(s) ds \\ & \text{sous } \begin{cases} x(0) = x_0, \quad x(N+1) = x_f, \\ x(i) \in G_{\sigma_i, \sigma_{i+1}}, & i \in \{1, \dots, N\}, \\ \forall s \in [i, i+1], x(s) \in D_{\sigma_i}, & i \in \{0, \dots, N\}, \end{cases} \end{aligned} \quad (9.0.1)$$

$$\text{avec } \forall s \in [i, i+1], x'(s) = \frac{1}{t_{i+1} - t_i} (A_{\sigma_i} x(s) + B_{\sigma_i} u(s)), \quad i \in \{0, \dots, N\}.$$

L'approche que nous souhaitons utiliser pour le calcul d'une solution approchée de ce problème est la suivante (voir aussi [50]). Sur chaque intervalle $[i, i+1]$, on peut exprimer l'entrée du système hybride dans la base d'ondelettes de la théorie du contrôle associée au couple de matrices $(A_{\sigma_i}, B_{\sigma_i})$.

Pour calculer une approximation de l'entrée optimale, on discrétise, dans un premier temps, les contraintes d'état :

$$\forall s \in [i, i+1], x(s) \in D_{\sigma_i}, \quad i \in \{0, \dots, N\},$$

en les remplaçant par

$$x(i + k/2^J) \in D_{\sigma_i}, \quad i \in \{0, \dots, N\}, \quad k \in \{1, \dots, 2^J\}$$

avec $J \in \mathbb{N}$.

Nous verrons que u_J , la solution exacte de la version discrétisée du problème peut être exprimée comme une combinaison linéaire des premières fonctions de la base d'ondelettes (aux échelles inférieures à J).

La sortie associée x_J , est généralement une bonne approximation de la sortie optimale du problème (9.0.1). Cependant, elle ne respecte pas toutes les contraintes de (9.0.1). Cette sortie n'est donc pas une exécution acceptée par le système hybride.

On procède alors à une deuxième phase, que l'on nomme régularisation, il s'agit de modifier légèrement l'entrée solution du problème discrétisé, pour que l'exécution associée soit effectivement acceptée par le système hybride. Pour rester proche de x_J , on ne modifie pas la valeur de la sortie aux points de la discrétisation; on se contente de la modifier localement entre deux de ces points. Nous verrons que ceci peut se faire en ajoutant à l'entrée u_J des coefficients d'ondelettes aux échelles supérieures à J .

Pour ne pas alourdir inutilement des notations déjà chargées, nous expliquerons la méthode sur le problème de contrôle optimal d'un système dynamique linéaire :

$$\begin{aligned} & \text{Minimiser } \int_0^1 u^T(t)u(t) ds \\ & \text{sous } \begin{cases} x(0) = x_0, x(1) = x_f \\ \forall t \in [0, 1], c^T x(t) - d \geq 0, \end{cases} \quad (9.0.2) \\ & \text{avec } x'(t) = Ax(t) + Bu(t). \end{aligned}$$

où c est un vecteur de \mathbb{R}^n et d un réel.

9.1 Discrétisation du problème de contrôle optimal

Soit $J \in \mathbb{N}$, la discrétisation des contraintes du problème (9.0.2) nous mène à :

$$\begin{aligned} & \text{Minimiser } \int_0^1 u^T(t)u(t) ds \\ & \text{sous } \begin{cases} x(0) = x_0, x(1) = x_f \\ c^T x(k/2^J) - d \geq 0, k \in \{1, \dots, 2^J\}, \end{cases} \quad (9.1.3) \\ & \text{avec } x'(t) = Ax(t) + Bu(t). \end{aligned}$$

On a vu, dans le chapitre précédent, que la solution d'un problème de contrôle optimal pouvait être recherchée dans le sous-espace $\mathcal{O}_{(A,B)}$.

Ainsi, on peut remplacer u dans (9.1.3) par sa décomposition dans la base d'ondelettes de la théorie du contrôle :

$$u = \sum_{l=1}^{l=n} \alpha^l \varphi^l + \sum_{j=0}^{j=\infty} \sum_{m=2^j-1}^{m=2^{j+1}-1} \sum_{l=1}^{l=n} \beta_{j,m}^l \psi_{j,m}^l.$$

D'après la proposition 8.2.1,

$$\forall t \in [0, 1], x(t) = e^{At}x_0 + \sum_{l=1}^{l=n} \alpha^l \chi^l(t) + \sum_{j=0}^{j=\infty} \sum_{m=2^j-1}^{m=2^{j+1}-1} \sum_{l=1}^{l=n} \beta_{j,m}^l \xi_{j,m}^l(t).$$

D'après la propriété de hiérarchisation de la proposition 8.2.2, aux points de discrétisation, la valeur de la sortie x ne dépend que d'un nombre fini de coefficients de la décomposition

de l'entrée u dans la base d'ondelettes. Pour tout $k \in \{1, \dots, 2^J\}$,

$$x(k/2^J) = e^{Ak/2^J} x_0 + \sum_{l=1}^{l=n} \alpha^l \chi^l(k/2^J) + \sum_{j=0}^{j=J-1} \sum_{m=0}^{m=2^j-1} \sum_{l=1}^{l=n} \beta_{j,m}^l \xi_{j,m}^l(k/2^J).$$

Ainsi, les coefficients de la décomposition en ondelettes

$$\{\beta_{j,m}^l, j \geq J, m \in \{0, \dots, 2^j - 1\}, l \in \{1, \dots, n\}\}$$

n'ont pas d'incidence sur la satisfaction des contraintes du problème (9.1.3). De plus, la base d'ondelettes étant orthonormale, on a

$$\int_0^1 u^T(t)u(t) ds = \sum_{l=1}^{l=n} (\alpha^l)^2 + \sum_{j=0}^{j=\infty} \sum_{m=0}^{m=2^j-1} \sum_{l=1}^{l=n} (\beta_{j,m}^l)^2.$$

Il apparait donc clairement que la solution u_J de (9.1.3) est de la forme :

$$u_J = \sum_{l=1}^{l=n} \alpha^l \varphi^l + \sum_{j=0}^{j=J-1} \sum_{m=0}^{m=2^j-1} \sum_{l=1}^{l=n} \beta_{j,m}^l \psi_{j,m}^l.$$

Le problème (9.1.3) a donc sa solution dans V_J , qui est de dimension finie. Le problème de contrôle optimal se ramène donc à un problème de programmation linéaire quadratique¹. Pour mieux visualiser la structure du problème, on renumérote les points de discrétisation :

$$t_{i,p} = \frac{1}{2^{i+1}} + \frac{p}{2^i}, \quad i \in \{0, \dots, J-1\}, \quad p \in \{0, \dots, 2^i - 1\}.$$

Lemme 9.1.1 Soit x_J la sortie associée à l'entrée u_J :

$$\begin{aligned} x_J(1) &= e^A x_0 + \sum_{l=1}^{l=n} \alpha^l \chi^l(1), \\ x_J(t_{i,p}) &= e^{At_{i,p}} x_0 + \sum_{l=1}^{l=n} \alpha^l \chi^l(t_{i,p}) + \sum_{j=0}^{j=i} \sum_{l=1}^{l=n} \beta_{j, [2^{j-i}p]}^l \xi_{j, [2^{j-i}p]}^l(t_{i,p}). \end{aligned}$$

où $[2^{j-i}p]$ dénote la partie entière de $2^{j-i}p$.

Preuve : D'après la propriété de hiérarchisation de la proposition 8.2.2, on a

$$\begin{aligned} x_J(1) &= e^A x_0 + \sum_{l=1}^{l=n} \alpha^l \chi^l(1), \\ x_J(t_{i,p}) &= e^{At_{i,p}} x_0 + \sum_{l=1}^{l=n} \alpha^l \chi^l(t_{i,p}) + \sum_{j=0}^{j=i} \sum_{m=0}^{m=2^j-1} \sum_{l=1}^{l=n} \beta_{j,m}^l \xi_{j,m}^l(t_{i,p}). \end{aligned}$$

D'après la propriété de localisation de la proposition 8.2.2, pour $j \leq i$,

$$\begin{aligned} \xi_{j,m}^l(t_{i,p}) \neq 0 &\implies m/2^j < t_{i,p} < (m+1)/2^j \\ &\implies m < \frac{p}{2^{i-j}} + \frac{1}{2^{i-j+1}} < m+1 \\ &\implies m = [2^{j-i}p]. \end{aligned}$$

En reportant dans l'expression de $x_J(t_{i,p})$ on arrive à l'expression voulue.

¹Minimisation sur un espace vectoriel de dimension finie d'une forme quadratique sous des contraintes linéaires [94].

■

On peut donc reformuler le problème de contrôle optimal (9.1.3) d'une manière équivalente :

$$\begin{aligned} & \text{Minimiser} \quad \sum_{l=1}^{l=n} (\alpha^l)^2 + \sum_{j=0}^{j=J-1} \sum_{m=0}^{m=2^j-1} \sum_{l=1}^{l=n} (\beta_{j,m}^l)^2 \\ & \quad \alpha^l, \beta_{j,m}^l \\ \text{sous} \quad & \begin{cases} \sum_{l=1}^{l=n} \alpha^l \chi^l(1) & = x_f - e^A x_0 \\ \sum_{l=1}^{l=n} \alpha^l c^T \chi^l(t_{i,p}) + \sum_{j=0}^{j=i} \sum_{l=1}^{l=n} \beta_{j,[2^j-i-p]}^l c^T \xi_{j,[2^j-i-p]}^l(t_{i,p}) & \geq d - c^T e^{A t_{i,p}} x_0, \\ i \in \{0, \dots, J-1\}, p \in \{0, \dots, 2^i-1\}. \end{cases} \end{aligned} \quad (9.1.4)$$

Ainsi, la solution de la version discrétisée du problème de contrôle optimal peut être calculée en résolvant un problème de programmation linéaire quadratique avec $2^J n$ variables, n contraintes égalités et $2^J - 1$ contraintes inégalités.

Pour la satisfaction de la contrainte à l'instant $t_{i,p}$, uniquement $n + n(i+1)$ coefficients de la décomposition sont impliqués. En écrivant les contraintes inégalités sous forme matricielle, le nombre de coefficients non-nuls de la matrices de contraintes est donc

$$\sum_{i=0}^{i=J-1} \sum_{p=0}^{p=2^i-1} n(i+2) = \sum_{i=0}^{i=J-1} n(i+2)2^i = J2^J n.$$

Cette matrice de dimension $(2^J - 1) \times 2^J n$ est donc très creuse. Le problème peut alors être résolu de manière efficace en utilisant des solveurs adaptés [94].

9.2 Régularisation de la sortie

Nous avons donc un moyen de calculer une entrée u_J telle que la sortie associée x_J vérifie les contraintes du problème (9.0.2) aux points de discrétisation.

Pour J suffisamment grand, u_J est généralement une bonne approximation de la solution de (9.0.2). Cependant, la sortie associée ne satisfait généralement pas les contraintes sur l'intervalle $[0, 1]$.

Dans cette section, nous expliquons comment régulariser cette sortie pour satisfaire les contraintes sur $[0, 1]$. Pour cela, on ajoute à l'entrée u_J , une seconde entrée v_J , telle que la sortie y_J associée à l'entrée $u_J + v_J$ garde la même allure que x_J (y_J interpole x_J aux points de discrétisation) mais telle que y_J satisfasse les contraintes sur $[0, 1]$.

D'après la propriété de hiérarchisation de la proposition 8.2.2, pour que y_J ait la même valeur que x_J aux points de discrétisation, il suffit de rechercher v_J sous la forme

$$v_J = \sum_{j=J}^{j=\infty} \sum_{m=0}^{m=2^j-1} \sum_{l=1}^{l=n} \beta_{j,m}^l \psi_{j,m}^l.$$

Il est bien évidemment nécessaire que v_J s'exprime en fonction d'un nombre fini de coefficients, on le cherche donc sous la forme

$$v_J = \sum_{m=0}^{m=2^J-1} \sum_{l=1}^{l=n} \beta_{J,m}^l \psi_{J,m}^l.$$

La sortie associée à l'entrée $u_J + v_J$ est alors donnée par

$$\forall t \in [0, 1], y_J(t) = x_J(t) + \sum_{m=0}^{m=2^J-1} \sum_{l=1}^{l=n} \beta_{J,m}^l \xi_{J,m}^l(t).$$

D'après la propriété de localisation de la proposition 8.2.2, pour tout $m \in \{0, \dots, 2^J - 1\}$,

$$\forall t \in [m/2^J, (m+1)/2^J], y_J(t) = x_J(t) + \sum_{l=1}^{l=n} \beta_{J,m}^l \xi_{J,m}^l(t).$$

La régularisation peut donc se faire de manière indépendante sur chaque intervalle de la forme $[m/2^J, (m+1)/2^J]$.

Soit donc $m \in \{0, \dots, 2^J - 1\}$. Remarquons que sur tout intervalle ouvert $I_{J,m} =]m/2^J, (m+1)/2^J[$, l'entrée u_J est de classe C^∞ et donc la sortie x_J est également de classe C^∞ . Sa dérivée seconde est donc bornée sur $I_{J,m}$. De même, sur les intervalles ouverts

$$I_{J+1,2m} =]m/2^J, (2m+1)/2^{J+1}[, I_{J+1,2m+1} =](2m+1)/2^{J+1}, (m+1)/2^J[,$$

les fonctions de la base induite des sorties $\{\xi_{J,m}^l, l \in \{1, \dots, n\}\}$ sont de classe C^∞ . Leur dérivées secondes sont donc bornées sur $I_{J+1,2m}$ et $I_{J+1,2m+1}$.

En utilisant la formule de Taylor à l'ordre 2, on a sur l'intervalle $I_{J+1,2m}$:

$$\begin{aligned} c^T y_J(t) &\geq c^T x_J\left(\frac{m}{2^J}\right) + \left(c^T x_J'\left(\frac{m}{2^J}\right) + \sum_{l=1}^{l=n} \beta_{J,m}^l c^T \xi_{J,m}^l\left(\frac{m}{2^J}\right) \right) (t - m/2^J) \\ &\quad - \left(\sup_{I_{J+1,2m}} |c^T x_J''| + \sum_{l=1}^{l=n} |\beta_{J,m}^l| \sup_{I_{J+1,2m}} |c^T \xi_{J,m}^{\prime\prime l}| \right) (t - m/2^J)^2/2. \end{aligned}$$

L'idée de la phase de régularisation est la suivante : si le second membre de l'inégalité ci-dessus est supérieur à d sur $I_{J+1,2m}$, alors y_J satisfait les contraintes sur l'intervalle $I_{J+1,2m}$. Le second membre de l'inégalité est un polynôme du second degré dont le terme quadratique est négatif, il est donc concave. A l'instant $t = m/2^J$, il est supérieur à d ; donc il est supérieur à d sur l'intervalle $I_{J+1,2m}$ si et seulement si il est supérieur à d à l'instant $t = (2m+1)/2^{J+1}$. Par conséquent,

$$\begin{aligned} d &\leq c^T x_J\left(\frac{m}{2^J}\right) + \left(c^T x_J'\left(\frac{m}{2^J}\right) + \sum_{l=1}^{l=n} \beta_{J,m}^l c^T \xi_{J,m}^l\left(\frac{m}{2^J}\right) \right) 2^{-J-1} \\ &\quad - \left(\sup_{I_{J+1,2m}} |c^T x_J''| + \sum_{l=1}^{l=n} |\beta_{J,m}^l| \sup_{I_{J+1,2m}} |c^T \xi_{J,m}^{\prime\prime l}| \right) 2^{-2J-3} \end{aligned}$$

est suffisant pour assurer que $c^T y_J(t) \geq d$ sur l'intervalle $I_{J+1,2m}$. Par des considérations similaires, on peut montrer que

$$\begin{aligned} d &\leq c^T x_J\left(\frac{m+1}{2^J}\right) - \left(c^T x_J'\left(\frac{m+1}{2^J}\right) + \sum_{l=1}^{l=n} \beta_{J,m}^l c^T \xi_{J,m}^l\left(\frac{m+1}{2^J}\right) \right) 2^{-J-1} \\ &\quad - \left(\sup_{I_{J+1,2m+1}} |c^T x_J''| + \sum_{l=1}^{l=n} |\beta_{J,m}^l| \sup_{I_{J+1,2m+1}} |c^T \xi_{J,m}^{\prime\prime l}| \right) 2^{-2J-3} \end{aligned}$$

est suffisant pour assurer que $c^T y_J(t) \geq d$ sur l'intervalle $I_{J+1,2m+1}$.

Ainsi, on choisit l'entrée v_J la plus petite possible, mais telle que les inégalités précédentes soient satisfaites. Donc, pour $m \in \{0, \dots, 2^J - 1\}$, les coefficients $\{\beta_{J,m}^l, l \in \{1, \dots, n\}\}$ doivent être solutions du problème d'optimisation :

$$\begin{aligned} & \text{Minimiser} \quad \sum_{l=1}^{l=n} (\beta_{J,m}^l)^2 \\ & \beta_{J,m}^l \\ \text{sous} \quad & \begin{cases} \sum_{l=1}^{l=n} \beta_{J,m}^l c^T \xi_{J,m}^{\prime l}(\frac{m}{2^J}) - |\beta_{J,m}^l| \sup_{I_{J+1,2m}} |c^T \xi_{J,m}^{\prime\prime l}| 2^{-J-2} \geq d_{J+1,2m} \\ \sum_{l=1}^{l=n} -\beta_{J,m}^l c^T \xi_{J,m}^{\prime l}(\frac{m+1}{2^J}) - |\beta_{J,m}^l| \sup_{I_{J+1,2m+1}} |c^T \xi_{J,m}^{\prime\prime l}| 2^{-J-2} \geq d_{J+1,2m+1} \end{cases} \end{aligned} \quad (9.2.5)$$

où

$$\begin{aligned} d_{J+1,2m} &= 2^{J+1} \left(d - c^T x_J(\frac{m}{2^J}) \right) - c^T x'_J(\frac{m}{2^J}) + 2^{-N-2} \sup_{I_{J+1,2m}} |c^T \xi_{J,m}^{\prime\prime l}| \\ d_{J+1,2m+1} &= 2^{J+1} \left(d - c^T x_J(\frac{m+1}{2^J}) \right) + c^T x'_J(\frac{m+1}{2^J}) + 2^{-N-2} \sup_{I_{J+1,2m+1}} |c^T \xi_{J,m}^{\prime\prime l}|. \end{aligned}$$

En utilisant, une astuce classique de la théorie de l'optimisation, on introduit les variables auxiliaires $\gamma_{J,m}^l = |\beta_{J,m}^l|$. Le problème (9.2.5) est alors équivalent au problème de programmation linéaire quadratique :

$$\begin{aligned} & \text{Minimiser} \quad \sum_{l=1}^{l=n} (\gamma_{J,m}^l)^2 \\ & \beta_{J,m}^l, \gamma_{J,m}^l \\ \text{sous} \quad & \begin{cases} \sum_{l=1}^{l=n} \beta_{J,m}^l c^T \xi_{J,m}^{\prime l}(\frac{m}{2^J}) - \gamma_{J,m}^l \sup_{I_{J+1,2m}} |c^T \xi_{J,m}^{\prime\prime l}| 2^{-J-2} \geq d_{J+1,2m} \\ \sum_{l=1}^{l=n} -\beta_{J,m}^l c^T \xi_{J,m}^{\prime l}(\frac{m+1}{2^J}) - \gamma_{J,m}^l \sup_{I_{J+1,2m+1}} |c^T \xi_{J,m}^{\prime\prime l}| 2^{-J-2} \geq d_{J+1,2m+1} \\ \gamma_{J,m}^l - \beta_{J,m}^l \geq 0, l \in \{1, \dots, n\} \\ \gamma_{J,m}^l + \beta_{J,m}^l \geq 0, l \in \{1, \dots, n\}. \end{cases} \end{aligned} \quad (9.2.6)$$

Soit

$$v_J = \sum_{m=0}^{m=2^J-1} \sum_{l=1}^{l=n} \beta_{J,m}^l \psi_{J,m}^l$$

où les coefficients $\beta_{J,m}^l$ ont été calculés en résolvant les 2^J problèmes de programmation linéaire quadratique (9.2.6), pour $m \in \{0, \dots, 2^J - 1\}$. On a alors,

Proposition 9.2.1 *Soit u l'entrée solution du problème de contrôle optimal(9.0.2). Soit u_J , l'entrée solution de la version discrétisée (9.1.3), soit x_J la sortie associée. Soient v_J l'entrée obtenue par régularisation de x_J et y_J la sortie associée à $u_J + v_J$. On a*

$$\begin{cases} y_J(0) = x_0, y_J(1) = x_f \\ \forall t \in [0, 1], c^T y_J(t) - d \geq 0. \end{cases}$$

On a de plus,

$$\int_0^1 u_J^T(t) u_J(t) dt \leq \int_0^1 u^T(t) u(t) dt \leq \int_0^1 u_J^T(t) u_J(t) dt + \int_0^1 v_J^T(t) v_J(t) dt.$$

Preuve : La première propriété est immédiate de par la construction de u_J et v_J .

La sortie associée à l'entrée u satisfait les contraintes de la version discrétisée (9.1.3), donc

$$\int_0^1 u_J^T(t)u_J(t) dt \leq \int_0^1 u^T(t)u(t).$$

D'autre part, y_J satisfait les contraintes du problème (9.0.2), donc

$$\int_0^1 u^T(t)u(t) dt \leq \int_0^1 (u_J(t) + v_J(t))^T(u_J(t) + v_J(t)) dt.$$

Par construction, u_J et v_J sont orthogonales, donc

$$\int_0^1 (u_J(t) + v_J(t))^T(u_J(t) + v_J(t)) dt = \int_0^1 u_J^T(t)u_J(t) dt + \int_0^1 v_J^T(t)v_J(t) dt.$$

■

Remarque 9.2.1 *Il est possible que le problème de programmation linéaire quadratique (9.2.6) n'ait pas de solution. La méthode de régularisation exposée ici échoue. Pour y remédier, on peut, par exemple, ajouter à v_J des coefficients aux échelles supérieures à $J + 1$.*

9.3 Exemples

Nous avons implémenté cette méthode dans le langage scilab. Nous présentons ici deux exemples d'utilisation.

9.3.1 Résultats expérimentaux

On considère le problème de contrôle optimal scalaire :

$$\begin{aligned} & \text{Minimiser } \int_0^1 u^2(t) ds \\ & \text{sous } \begin{cases} x(0) = 1, x(1) = 0.8, \\ \forall t \in [0, 1], x(t) \geq 0.8, \end{cases} \\ & \text{avec } x'(t) = -x(t) + u(t). \end{aligned}$$

Sur la figure 9.1, nous avons tracé, pour $J = 3$, l'entrée u_3 solution de la version discrétisée du problème de contrôle optimal (avec un pas de discrétisation de 1/8). En bas à gauche, nous avons représenté la sortie x_3 associée à l'entrée u_3 . On peut voir qu'entre deux points de la discrétisation la valeur x_3 est parfois inférieure à 0.8.

L'entrée v_3 obtenue par régularisation de x_3 est tracée en haut à gauche, il convient de noter que la différence d'échelle entre la courbe de u_3 et celle de v_3 est de l'ordre de 1/10. La sortie y_3 associée à $u_3 + v_3$ est représentée en bas à droite. On peut remarquer, qu'aux points de la discrétisation la valeur de y_3 est la même que celle de x_3 et qu'entre ces points la valeur de y_3 est toujours supérieure à 0.8.

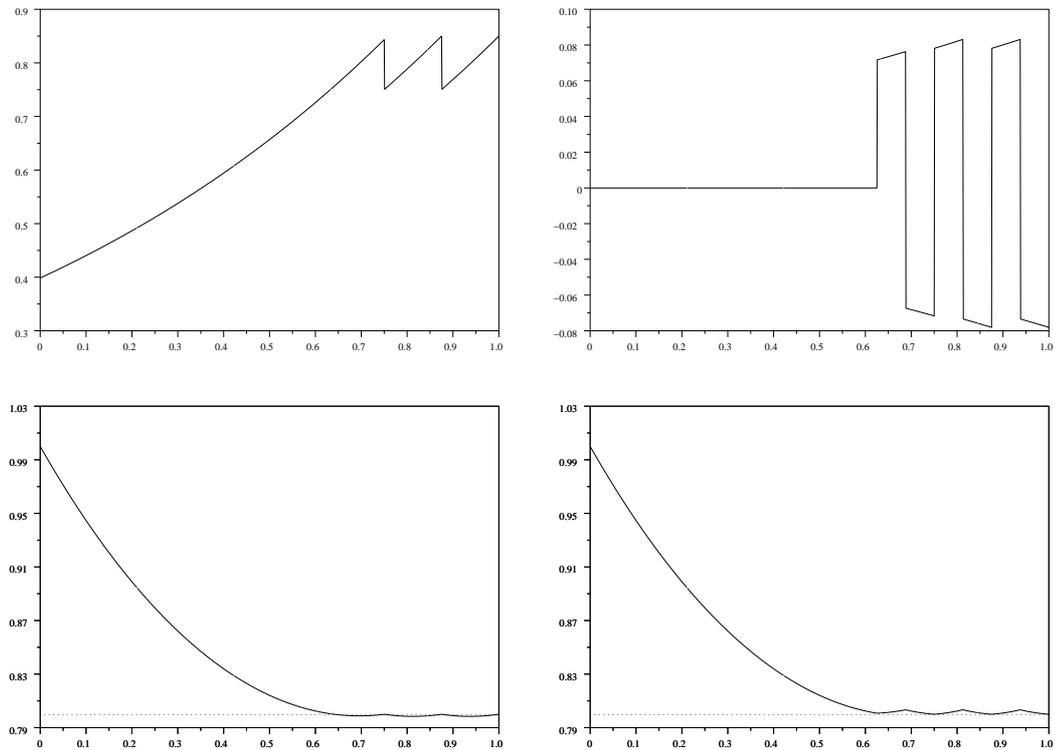


FIG. 9.1: Les entrées u_3 , (en haut à gauche), v_3 (en haut à droite) et les sorties x_3 (en bas à gauche), y_3 (en bas à droite).

Les résultats des expériences menées sur cet exemple sont répertoriés dans le tableau 9.1. Pour estimer la qualité de l'approximation (où l'erreur produite), on évalue $\|u_J + v_J\| - \|u_J\|$. Expérimentalement, il semble que la convergence soit de l'ordre $O(2^{-2J})$, c'est à dire proportionnel au carré du pas de discrétisation.

On voit également que le temps de calcul augmente relativement vite. Cependant, même pour des valeurs de J faibles, l'approximation est de relativement bonne qualité.

J	3	4	5	6	7	8
Erreur	1.6e-3	3.5e-4	8.5e-5	2.1e-5	5.2e-6	1.3e-6
Temps de calcul (s)	0	0.01	0.02	0.04	0.24	3.58

TAB. 9.1: Qualité de l'approximation, temps de calcul (Pentium III 1 GHz, Scilab).

9.3.2 Contrôle optimal d'un système hybride

La méthode présentée dans ce chapitre, s'étend facilement au contrôle des systèmes hybrides linéaires par morceaux.

Globalement, seule la première phase de la méthode change légèrement. Ainsi, on calcule la solution d'une version discrétisée de (9.0.1) :

$$\begin{aligned} & \text{Minimiser } \sum_{i=0}^{i=N} \frac{1}{t_{i+1}-t_i} \int_i^{i+1} u^T(s)u(s) ds \\ \text{sous } & \begin{cases} x(0) = x_0, x(N+1) = x_f, \\ x(i) \in G_{\sigma_i, \sigma_{i+1}}, & i \in \{1, \dots, N\}, \\ x(i + k/2^J) \in D_{\sigma_i}, k \in \{1, \dots, 2^J\}, & i \in \{0, \dots, N\}, \end{cases} \end{aligned} \quad (9.3.7)$$

$$\text{avec } \forall s \in [i, i+1], x'(s) = \frac{1}{t_{i+1}-t_i} (A_{\sigma_i}x(s) + B_{\sigma_i}u(s)), i \in \{0, \dots, N\}.$$

Ce problème peut être résolu en utilisant les bases d'ondelettes de contrôles associées à chaque mode d'une manière similaire à celle présentée ici. Le problème discrétisé peut alors se transformer en un problème de programmation linéaire quadratique.

La phase de régularisation est en tout point identique à celle présentée précédemment.

Nous avons utilisé notre méthode pour calculer une approximation de la solution du problème :

$$\begin{aligned} & \text{Minimiser } \int_0^2 u^T(t)u(t)dt \\ \text{sous } & \begin{cases} x^T(0) = -(1 \ 1), x^T(2) = (1 \ 1) \\ (0 \ 1).x(1) - 0.5 = 0 \\ (1 \ 0).x(t) + 1.5 \geq 0, \text{ si } t \in [0, 1] \\ (0 \ 1).x(t) - 0.5 \geq 0, \text{ si } t \in [1, 2]. \end{cases} \end{aligned}$$

avec

$$x'(t) = \begin{cases} A_1x(t) + B_1u(t), \text{ for } t \in [0, 1] \\ A_2x(t) + B_2u(t), \text{ for } t \in [1, 2] \end{cases}$$

où

$$A_1 = \begin{pmatrix} -0.1 & 0 \\ 0.5 & -1 \end{pmatrix}, \quad B_1 = \begin{pmatrix} 1 & 0 \\ -0.1 & 0.2 \end{pmatrix},$$

$$A_2 = \begin{pmatrix} -0.1 & -0.5 \\ 0.5 & -0.1 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 0.8 & 0 \\ 0.2 & 1 \end{pmatrix}.$$

La méthode a été utilisée avec un pas de discrétisation 1/32 ($J = 5$). Sur la figure 9.2, nous avons représenté les deux composantes de l'entrée $u_5 + v_5$ que nous avons calculée (en haut) ainsi que la variable continue y_5 de l'exécution associée à cette entrée.

En voit notamment, que cette exécution est effectivement acceptée par le système hybride : on ne sort pas des domaines entre deux transitions.

Nous avons également tracé, en bas, les deux composantes de l'entrée optimale.

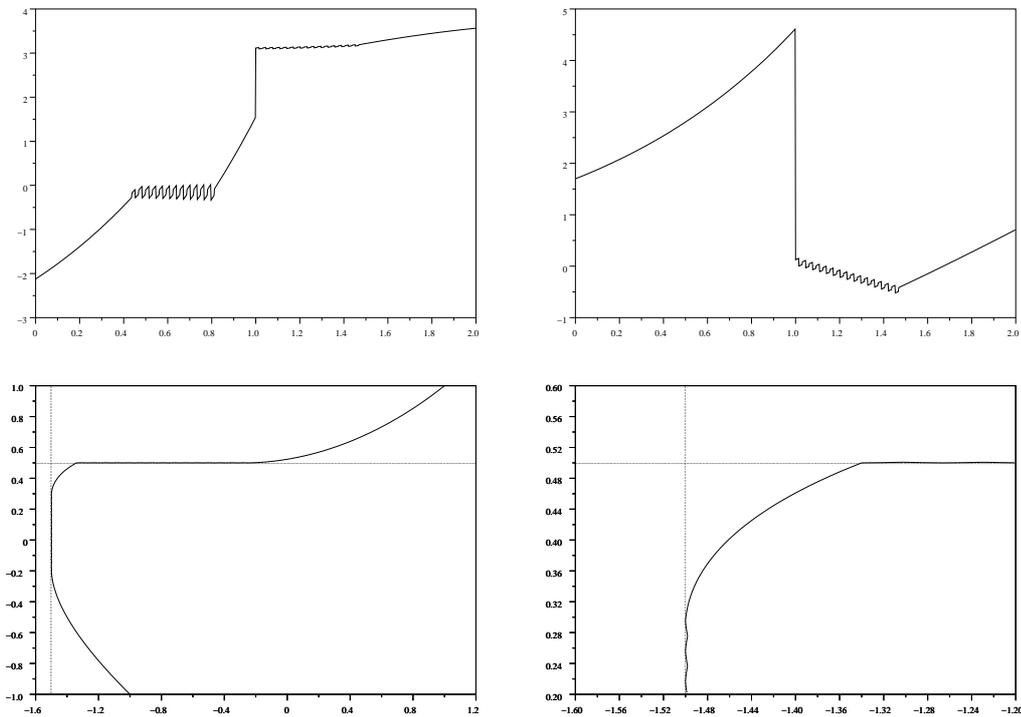


FIG. 9.2: L'entrée optimale $u_5 + v_5$ (en haut) et la sortie associée y_5 (en bas).

La norme L^2 de l'entrée u_5 est 4.5864 alors que celle de $u_5 + v_5$ est 4.5876. Le surcoût de l'entrée produite par notre méthode, par rapport à la véritable entrée optimale est de l'ordre de 0.03 pourcent, ce qui est négligeable.

Dans cette partie, nous avons donc développé une approche multirésolution pour résoudre des problèmes de contrôle de systèmes linéaires.

Il en résulte une analyse multirésolution, ainsi que la notion d'ondelettes de la théorie du contrôle.

Nous avons également proposé une méthodologie pour le calcul d'une approximation de l'entrée optimale d'un problème de contrôle d'un système hybride. L'exécution associée à l'entrée calculée, est effectivement une exécution acceptée par le système hybride.

Cinquième partie

**Approximation de Dynamiques
Non-Linéaires**

Chapitre 10

Calcul Hybride : Equation Différentielle Scalaire

La notion de calcul hybride a été introduite par Della Dora, Maignan, Mirica-Ruse et Yovine dans [28]. L'idée du calcul hybride est d'importer les techniques innovantes de la théorie des systèmes hybrides et de les appliquer à des problèmes, plus classiques, issus des mathématiques appliquées.

Un aperçu des différents problèmes qui peuvent être traités par le calcul hybride peut être trouvé dans la thèse de M. Mirica-Ruse [74]. On peut citer, entre autres, des problèmes de design géométrique, de contrôle optimal ou de résolution des équations différentielles ordinaires.

Considérons une équation différentielle, $x'(t) = f(x(t))$. La méthode d'approximation des solutions, par le calcul hybride, est la suivante. On se donne une partition de l'espace de phase. Dans chaque élément de cette partition, on calcule une approximation locale du champ de vecteurs f . La partition de l'espace, munie de la dynamique continue approchée, définit un système hybride, que l'on peut alors étudier.

Le succès de cette méthode réside dans le choix des approximations locales du champ de vecteurs; celles-ci doivent être suffisamment simples pour permettre l'étude algorithmique mais suffisamment riches pour garder une approximation de bonne qualité. Nous verrons que le choix d'approximations linéaires est un bon compromis.

Dans ce chapitre, nous appliquons les idées du calcul hybride à la résolution d'équations différentielles autonomes scalaires [45].

10.1 Exemple introductif

Expliquons l'idée en considérant un exemple relativement simple :

$$x'(t) = (x(t))^2, \quad x(0) = 1.$$

La solution de ce problème de Cauchy est $x(t) = 1/(1-t)$. Nous allons approcher $x(t)$ par $x_h(t)$, l'application continue d'une exécution d'un système hybride linéaire par morceaux.

Un point important du calcul hybride est que l'on a une discrétisation de l'espace des phases (ici \mathbb{R}); contrairement aux méthodes classiques d'approximation des solutions d'équations différentielles qui utilisent une discrétisation du temps (voir par exemple [29, 32]).

Soit h un réel positif, l'axe des réels est découpé en intervalles de longueur h , $D_i = [ih, (i+1)h]$. Chaque intervalle de la partition correspond au domaine d'un des états discrets du système hybride.

Sur chaque intervalle D_i , on définit une fonction f_i , approximation locale du champ de vecteurs $f : x \mapsto x^2$. On peut, par exemple, prendre l'interpolant linéaire de f aux points ih et $(i+1)h$:

$$f_i(x) = a_i x + b_i$$

avec

$$a_i = \frac{f((i+1)h) - f(ih)}{h} \text{ et } b_i = f(ih) - a_i(ih),$$

soit

$$a_i = (2i+1)h \text{ et } b_i = -i(i+1)h^2.$$

Soit $x_h(t)$ la solution de l'équation différentielle linéaire par morceaux :

$$x'_h(t) = f_i(x_h(t)), \text{ si } x_h(t) \in D_i, x_h(0) = 1.$$

Il est facile de montrer que l'on a

$$x_h(t) = \left(ih + \frac{b_i}{a_i}\right)e^{a_i(t-t_i)} - \frac{b_i}{a_i}, \text{ si } x_h(t) \in D_i$$

où t_i est la valeur de l'instant où $x_h(t_i) = ih$. On peut alors calculer l'instant t_{i+1} de la transition suivante, qui survient lorsque $x_h(t_{i+1}) = (i+1)h$.

$$t_{i+1} - t_i = \frac{2}{(2i+1)h} \ln\left(\frac{i+1}{i}\right).$$

On peut voir ici, une des principales différences avec les méthodes classiques; la subdivision du temps est définie implicitement, par la définition de l'approximation linéaire par morceaux, et n'est pas fixée a priori.

Remarquons que la solution de l'équation différentielle initiale $x(t)$ possède une asymptote verticale en $t = 1$. L'existence d'une asymptote pour $x_h(t)$, signifierait qu'une infinité de domaines D_i sont parcourus par $x_h(t)$ en temps fini. Le système hybride sous-jacent à l'équation différentielle linéaire par morceaux effectue donc une infinité de transitions en temps fini, et donc, l'exécution acceptée par le système hybride est Zénon.

Prenons $h = 1/N$ avec $N \in \mathbb{N}^*$. On a $x_h(0) = 1 = Nh$, et donc $t_N = 0$. L'existence d'une asymptote verticale pour $x_h(t)$ se traduit par

$$\lim_{i \rightarrow \infty} t_i = \sum_{i \geq N} (t_{i+1} - t_i) < +\infty.$$

Il est facile de montrer que

$$2N \int_N^\infty \frac{1}{2x+1} \ln\left(\frac{x+1}{x}\right) dx \leq \sum_{i \geq N} (t_{i+1} - t_i) \leq 2N \int_N^\infty \frac{1}{2x-1} \ln\left(\frac{x}{x-1}\right) dx.$$

Ces intégrales sont finies et, de plus, leur limite quand N tend vers l'infini est 1. D'où,

$$\lim_{N \rightarrow \infty} \sum_{i \geq N} (t_{i+1} - t_i) = 1.$$

$x_h(t)$ possède donc une asymptote verticale qui tend vers 1 quand h tend vers 0.

Sur la figure 10.1, on a représenté $x(t)$ et $x_h(t)$ pour différentes valeurs de h . Expérimentalement, on trouve que l'ordre d'approximation de $x(t)$ par $x_h(t)$ est de l'ordre $O(h^2)$.

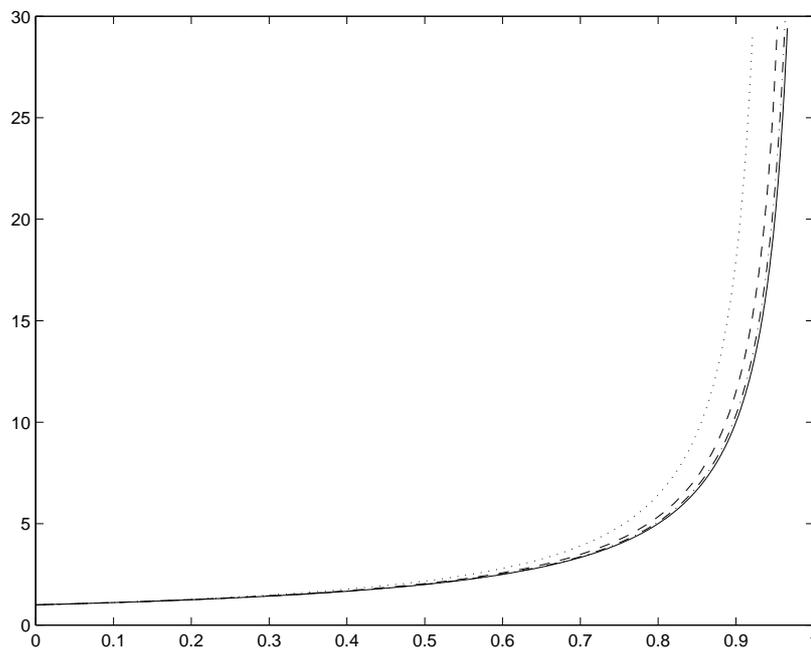


FIG. 10.1: $x_h(t)$ pour $h \in \{0.125, 0.25, 0.5, 1\}$ et $x(t)$ (en trait plein)

10.2 Approximation des solutions d'équations différentielles

Dans cette partie, nous généralisons la méthode d'approximation décrite pour l'exemple $x' = x^2$.

10.2.1 Méthode générale

Nous considérons des équations différentielles scalaires. Soit

$$x'(t) = f(x(t)), \quad x(t) \in [a, b], \quad x(0) = x_0. \quad (10.2.1)$$

Si $f(x_0) = 0$ alors pour tout t , $x(t) = x_0$. On suppose donc que $f(x_0) > 0$, le cas $f(x_0) < 0$ est symétrique. Ainsi, pour tout $t \geq 0$, $f(x(t)) > 0$.

Première étape : approximation de l'équation différentielle

On définit une partition uniforme $(D_i)_{i \in I}$, de pas h , de l'intervalle $[a, b]$. Sur chaque intervalle $D_i = [\nu_i, \nu_{i+1}]$, on note f_i l'interpolant linéaire de f aux points ν_i et ν_{i+1} :

$$f_i(x) = a_i x + b_i$$

avec

$$a_i = \frac{f(\nu_{i+1}) - f(\nu_i)}{h}, \text{ et } b_i = f(\nu_i) - a_i \nu_i.$$

On définit ensuite l'approximation linéaire par morceaux de la fonction f :

$$f_h(x) = f_i(x), \text{ si } x \in D_i.$$

La méthode de calcul hybride revient à remplacer la résolution de (10.2.1) par celle de

$$x'_h(t) = f_h(x_h(t)), \quad x_h(t) \in [a, b], \quad x_h(0) = x_0. \quad (10.2.2)$$

Deuxième étape : résolution de l'équation différentielle approchée

Il est clair que, pour h suffisamment petit, $f_h(x_0) > 0$. Donc $x_h(t)$ est croissante. La solution de (10.2.2) est donc donnée par

$$x_h(t) = \begin{cases} \nu_i + b_i(t - t_i) & \text{si } f(\nu_i) = f(\nu_{i+1}) \\ (\nu_i + \frac{b_i}{a_i})e^{a_i(t-t_i)} - \frac{b_i}{a_i} & \text{sinon} \end{cases}, \text{ si } x_h(t) \in D_i$$

où t_i dénote l'instant auquel $x_h(t_i) = \nu_i$. Le calcul de $x_h(t)$ consiste donc, essentiellement (comme pour la simulation des systèmes hybrides d'ailleurs), à calculer la séquence des instants t_i , qui correspondent aux instants de transition d'une dynamique linéaire à une autre dynamique linéaire. La séquence des t_i est donnée par la relation de récurrence

$$t_{i+1} = \begin{cases} +\infty & \text{si } f(\nu_{i+1}) \leq 0 \\ t_i + \frac{h}{b_i} & \text{si } f(\nu_i) = f(\nu_{i+1}) \\ t_i + \frac{\ln(f(\nu_{i+1})) - \ln(f(\nu_i))}{a_i} & \text{sinon.} \end{cases}$$

10.2.2 Convergence de l'approximation

Le résultat de convergence de $x_h(t)$ vers $x(t)$ est une conséquence du théorème suivant.

Théorème 10.2.1 (Inégalité fondamentale [30]) *Soit f une application de \mathbb{R}^n dans \mathbb{R}^n , continue et L -Lipschitz sur un ensemble $D \subseteq \mathbb{R}^n$.*

Soient x_1 et x_2 deux fonctions à valeurs dans l'ensemble D , continues, dérivables par morceaux sur un intervalle $I \subseteq \mathbb{R}$ contenant 0, et vérifiant :

$$\forall t \in I \text{ où } x_i \text{ est dérivable, } \|x'_i(t) - f(x_i(t))\| \leq \varepsilon_i, \quad i \in \{1, 2\}.$$

Alors,

$$\forall t \in I, \|x_1(t) - x_2(t)\| \leq \|x_1(0) - x_2(0)\| e^{L|t|} + \frac{\varepsilon_1 + \varepsilon_2}{L} (e^{L|t|} - 1).$$

Appliquons ce théorème à notre problème d'approximation. Supposons que le champ de vecteurs f est L -Lipschitz et C^2 sur $[a, b]$. Il existe un intervalle I et une fonction $x(t)$ de I dans $[a, b]$, continue et dérivable, tels que :

$$\forall t \in I, |x'(t) - f(x(t))| = 0.$$

De même, il existe I_h et une fonction $x_h(t)$ de I_h dans $[a, b]$, continue et dérivable, telle que :

$$\forall t \in I_h, |x'_h(t) - f_h(x_h(t))| = 0.$$

On a, grâce à un résultat classique d'interpolation [29],

$$\forall x \in [a, b], |f(x) - f_h(x)| \leq \sup_{[a,b]} |f''| \frac{h^2}{8}.$$

D'où,

$$\begin{aligned} \forall t \in I_h, |x'_h(t) - f(x_h(t))| &\leq |x'_h(t) - f_h(x_h(t))| + |f_h(x_h(t)) - f(x_h(t))| \\ &\leq \sup_{[a,b]} |f''| \frac{h^2}{8}. \end{aligned}$$

En remarquant $|x_h(0) - x(0)| = 0$, on peut alors énoncer le résultat de convergence suivant.

Théorème 10.2.2 (Convergence de l'approximation) *Soit x la solution de l'équation différentielle (10.2.1), où le champ de vecteurs f est L -Lipschitz et C^2 sur $[a, b]$; soit x_h la solution de l'équation différentielle (10.2.2).*

Pour tout t , où $x(t)$ et $x_h(t)$ sont définies,

$$|x(t) - x_h(t)| \leq \sup_{[a,b]} |f''| \frac{h^2}{8L} (e^{L|t|} - 1).$$

On voit donc que l'approximation de $x(t)$ par $x_h(t)$ est de l'ordre $O(h^2)$. Ceci confirme ce que l'on avait observé, expérimentalement, sur l'exemple $x' = x^2$.

10.2.3 Résultats expérimentaux

La méthode d'approximation des solutions d'équations différentielles scalaires a été implémentée¹ en C++.

Nous présentons ici les performances de la méthode pour l'exemple $x' = x^2$, avec la condition initiale $x(0) = 1$. Nous avons calculé la valeur de $x_h(0.95)$ pour différentes valeurs du pas de discrétisation h . Les erreurs d'approximation, ainsi que les temps de calcul, sont répertoriés dans le tableau 10.1.

On vérifie, expérimentalement, que $x_h(t)$ approche $x(t)$ avec une précision d'ordre $O(h^2)$, ce qui concorde avec l'estimation théorique donnée par le théorème 10.2.2.

Notons, également, que le coût de l'algorithme est en $O(1/h)$, ce qui est tout à fait raisonnable, et comparable au coût des méthodes classiques d'approximation des solutions d'équations différentielles.

¹en collaboration avec Jean-Guillaume Dumas

valeur de h	Erreur à $t = 0.95$	Temp d'exécution (Secondes)
10^{-1}	$2.2397 \cdot 10^{-1}$	0
10^{-2}	$2.2221 \cdot 10^{-3}$	0.001952
10^{-3}	$2.2219 \cdot 10^{-5}$	0.020496
10^{-4}	$2.2219 \cdot 10^{-7}$	0.203008
10^{-5}	$2.2304 \cdot 10^{-9}$	2.01739

TAB. 10.1: Résultats expérimentaux pour $x' = x^2$, $x(0) = 1$ (Dec Alpha 500 MHz, C++)

10.2.4 Limites de la méthode

Il existe certains cas, où la méthode d'approximation a un mauvais comportement. On a bien la convergence en $O(h^2)$, pour tout t , vers la solution exacte de l'équation différentielle; cependant, le comportement de $x_h(t)$ et celui de $x(t)$, quand t tend vers $+\infty$ est différent, et ce pour tout h , aussi petit soit-il.

Ceci peut se produire, par exemple, quand $x(t)$, la solution de (10.2.1), possède une asymptote horizontale. Dans ce cas, il existe $x_\infty \in [a, b]$, tel que,

$$\lim_{t \rightarrow +\infty} x(t) = x_\infty.$$

On a, nécessairement, $f(x_\infty) = 0$. Cependant, il est possible que l'approximation linéaire par morceaux f_h ne possède pas de zéro dans le voisinage de x_∞ . Ce cas est illustré sur la figure 10.2.

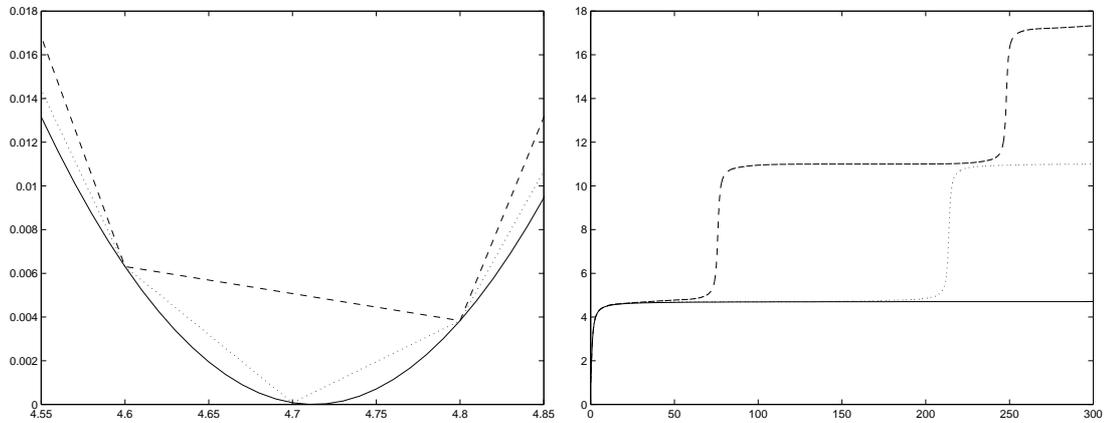


FIG. 10.2: Courbes représentatives de $1 + \sin(x)$ et ses approximations linéaires par morceaux (à gauche). Les approximations linéaires par morceaux ne s'annulent pas. Solutions de l'équation différentielle $x' = 1 + \sin(x)$ et des approximations linéaires par morceaux associées (à droite).

Considérons, l'équation différentielle

$$x'(t) = 1 + \sin(x(t)), \quad x(0) = 1.$$

La solution $x(t)$ a une asymptote horizontale en $x_\infty = 3\pi/2$. Cependant, il est visible sur la figure que l'approximation $x_h(t)$ n'en a pas.

Ce problème est lié au fait que le champ de vecteurs $f(x) = 1 + \sin(x)$ s'annule en $3\pi/2$, mais est strictement positif sur un voisinage de ce point. Ainsi, l'approximation linéaire par morceaux $f_h(x)$ de $f(x)$ ne peut s'annuler sur un voisinage de $3\pi/2$, sauf si le point $3\pi/2$ est un point d'interpolation.

La raison profonde de ce problème est que l'équation différentielle $x' = 1 + \sin(x)$ est très sensible aux perturbations. On dit que le champ de vecteurs $f(x) = 1 + \sin(x)$ est structurellement instable. Nous reparlerons de cette notion par la suite.

10.3 Améliorations de la convergence

Dans la section précédente, le choix de la fonction f_h était limité à l'interpolant linéaire par morceaux sur une subdivision uniforme de $[a, b]$. Cependant, il y a de nombreux autres choix possibles qui donnent de meilleurs résultats. Dans cette partie, nous présentons certains d'entre eux.

Tout d'abord, nous verrons qu'il est possible d'améliorer l'ordre de convergence en gardant des approximations linéaires par morceaux. Ensuite, nous verrons qu'il est également possible de prendre des approximations de degré supérieur.

10.3.1 Interpolation aux abscisses de Gauss

La première amélioration, dans le choix de l'approximation, a été inspirée par une remarque de Professeur P.J. Laurent. Pourquoi choisir l'interpolant aux points de discrétisation, alors qu'il est possible de choisir une approximation qui répartisse mieux l'erreur?

En effet, soit $x(t)$ la solution de l'équation différentielle (10.2.1). Comme $f(x_0) > 0$, $x(t)$ est strictement croissante. Considérons alors l'intégrale

$$t = \int_0^t ds.$$

On effectue dans l'intégrale, le changement de variable $x = x(s)$.

$$t = \int_{x_0}^{x(t)} \frac{1}{f(x)} dx.$$

Soit f_h une approximation de f , suffisamment précise pour que $f_h(x_0) > 0$. Soit $x_h(t)$ la solution de l'équation différentielle approchée (10.2.2) associée au champ de vecteurs f_h . On a également,

$$t = \int_{x_0}^{x_h(t)} \frac{1}{f_h(x)} dx.$$

D'où

$$\int_{x_h(t)}^{x(t)} \frac{1}{f(x)} dx = \int_{x_0}^{x_h(t)} \left(\frac{1}{f_h(x)} - \frac{1}{f(x)} \right) dx.$$

On a, par conséquent, une nouvelle majoration de l'erreur d'approximation :

$$|x(t) - x_h(t)| \leq \sup_{x \in [a, b]} |f(x)| \left| \int_{x_0}^{x_h(t)} \left(\frac{1}{f_h(x)} - \frac{1}{f(x)} \right) dx \right|. \quad (10.3.3)$$

On voit donc, que le meilleur choix, pour la fonction f_h , est la fonction qui minimise l'intégrale dans l'inégalité (10.3.3).

Soit $(D_i)_{i \in I}$ une partition de $[a, b]$, uniforme, de pas h . D'après la théorie de l'intégration numérique [29], l'intégrale sur l'intervalle D_i d'une fonction, suffisamment lisse, qui s'annule aux abscisses d'intégration de Gauss de D_i , est en $O(h^5)$. On choisit donc, pour minimiser l'intégrale,

$$f_h(x) = f_i(x) \text{ si } x \in D_i, \quad (10.3.4)$$

où f_i est l'interpolant linéaire de f aux abscisses d'intégration de Gauss, $\mu_{i,1}$ et $\mu_{i,2}$, de l'intervalle $D_i = [\nu_i, \nu_{i+1}]$:

$$\begin{aligned} \mu_{i,1} &= \frac{\sqrt{3}+1}{2\sqrt{3}}\nu_i + \frac{\sqrt{3}-1}{2\sqrt{3}}\nu_{i+1} \\ \mu_{i,2} &= \frac{\sqrt{3}-1}{2\sqrt{3}}\nu_i + \frac{\sqrt{3}+1}{2\sqrt{3}}\nu_{i+1}. \end{aligned}$$

Proposition 10.3.1 *Soit x la solution de l'équation différentielle (10.2.1), où le champ de vecteurs f est C^4 sur $[a, b]$; soit x_h la solution de l'équation différentielle (10.2.2), où le champ de vecteurs f_h est définie par (10.3.4).*

Pour tout t , où $x(t)$ et $x_h(t)$ sont définies,

$$|x(t) - x_h(t)| = O(h^3).$$

Preuve : Soient i_0 , l'indice de l'intervalle contenant x_0 , i_t l'indice de l'intervalle contenant $x_h(t)$. On a alors,

$$\begin{aligned} |x(t) - x_h(t)| &\leq \sup_{x \in [a, b]} |f(x)| \left| \sum_{i=i_0+1}^{i=i_t-1} \int_{\nu_i}^{\nu_{i+1}} \left(\frac{1}{f_h(x)} - \frac{1}{f(x)} \right) dx \right. \\ &\quad \left. + \int_{x_0}^{\nu_{i_0+1}} \left(\frac{1}{f_h(x)} - \frac{1}{f(x)} \right) dx + \int_{\nu_{i_t}}^{x_h(t)} \left(\frac{1}{f_h(x)} - \frac{1}{f(x)} \right) dx \right| \\ &\leq \sup_{x \in [a, b]} |f(x)| \left(\sum_{i=i_0+1}^{i=i_t-1} \int_{\nu_i}^{\nu_{i+1}} \left| \frac{1}{f_h(x)} - \frac{1}{f(x)} \right| dx \right. \\ &\quad \left. + \int_{x_0}^{\nu_{i_0+1}} \left| \frac{f(x) - f_h(x)}{f(x)f_h(x)} \right| dx + \int_{\nu_{i_t}}^{x_h(t)} \left| \frac{f(x) - f_h(x)}{f(x)f_h(x)} \right| dx \right) \quad (10.3.5) \end{aligned}$$

Comme f_h est un interpolant linéaire par morceaux de f sur la partition $(D_i)_{i \in I}$, on a, pour tout $x \in [a, b]$, $|f(x) - f_h(x)| = O(h^2)$, d'où :

$$\int_{x_0}^{\nu_{i_0+1}} \left| \frac{f(x) - f_h(x)}{f(x)f_h(x)} \right| dx + \int_{\nu_{i_t}}^{x_h(t)} \left| \frac{f(x) - f_h(x)}{f(x)f_h(x)} \right| dx = O(h^3).$$

Remarquons que, pour tout $i \in \{i_0 + 1, \dots, i_t - 1\}$, la fonction $\left| \frac{1}{f_h(x)} - \frac{1}{f(x)} \right|$ s'annule aux abscisses d'intégration de Gauss de l'intervalle D_i .

De plus, f est C^4 sur $[a, b]$, donc d'après la méthode d'intégration de Gauss [29],

$$\int_{\nu_i}^{\nu_{i+1}} \left| \frac{1}{f_h(x)} - \frac{1}{f(x)} \right| dx = O(h^5).$$

Finalement, en remarquant que $i_t - i_0 = O(1/h)$, on arrive à

$$|x(t) - x_h(t)| \leq \sup_{x \in [a,b]} |f(x)| \left(O(h^3) + \sum_{i=i_0+1}^{i=i_t-1} O(h^5) \right) = O(h^3).$$

■

Remarque 10.3.1 Si $x_0 = \nu_{i_0}$, alors pour tout $i \in \{i_0 + 1, \dots, i_t - 1\}$, on a

$$|x(t_i) - x_h(t_i)| = |x(t_i) - \nu_i| = O(h^4).$$

On voit donc qu'un choix judicieux des points d'interpolation peut augmenter l'ordre de la méthode. Une autre façon d'améliorer la convergence consiste à prendre des approximations d'ordre supérieure (quadratiques par morceaux par exemple).

10.3.2 Interpolation quadratique par morceaux

Le théorème 10.2.1 montre que l'ordre d'approximation des solutions d'une équation différentielle est au moins aussi bon que l'ordre d'approximation de la fonction f par f_h . Il est donc naturel de prendre des approximations plus précises que linéaires par morceaux. Nous considérons ici des approximations quadratiques par morceaux.

Soit $(D_i)_{i \in I}$ une partition uniforme, de pas h , de l'intervalle $[a, b]$. Sur chaque intervalle $D_i = [\nu_i, \nu_{i+1}]$, on note f_i l'interpolant quadratique de f aux points ν_i , $\nu_i + h/2$ et ν_{i+1} . On définit la fonction f_h comme l'approximation quadratique par morceaux de f :

$$f_h(x) = f_i(x) = a_i x^2 + b_i x + c_i \text{ si } x \in D_i. \quad (10.3.6)$$

La solution de l'équation différentielle associée au champ de vecteurs f_h est

$$x_h(t) = \begin{cases} \frac{\tan\left(\frac{t-t_i}{2}\sqrt{-\Delta_i} + \arctan\left(\frac{2a_i\nu_i+b_i}{\sqrt{-\Delta_i}}\right)\right)\sqrt{-\Delta_i}-b_i}{2a_i} & \text{si } \Delta_i = b_i^2 - 4a_i c_i < 0 \\ \frac{(t-t_i)r_i(r_i-\nu_i)+\nu_i}{(t-t_i)(r_i-\nu_i)+1} & \text{si } \Delta_i = 0 \text{ avec } r_i = \frac{-b_i}{2a_i} \\ \frac{-r_{i,1} + \frac{\nu_i-r_{i,1}}{\nu_i-r_{i,2}} r_{i,2} e^{(r_{i,1}-r_{i,2})(t-t_i)}}{-1 + \frac{\nu_i-r_{i,1}}{\nu_i-r_{i,2}} e^{(r_{i,1}-r_{i,2})(t-t_i)}} & \text{si } \Delta_i > 0 \text{ avec } r_{i,j} = \frac{-b_i+(-1)^j\sqrt{\Delta_i}}{2a_i} \end{cases}$$

où t_i est l'instant auquel $x_h(t_i) = \nu_i$. La séquence des instants t_i est donnée par les formules

$$t_{i+1} = t_i + \begin{cases} 2 \frac{\arctan\left(\frac{2a_i\nu_{i+1}+b_i}{\sqrt{-\Delta_i}}\right) - \arctan\left(\frac{2a_i\nu_i+b_i}{\sqrt{-\Delta_i}}\right)}{\sqrt{-\Delta_i}} & \text{si } \Delta_i = b_i^2 - 4a_i c_i < 0 \\ \frac{\nu_{i+1}-\nu_i}{(\nu_{i+1}-r_i)(\nu_i-r_i)} & \text{si } \Delta_i = 0 \text{ avec } r_i = \frac{-b_i}{2a_i} \\ \frac{\ln\left(\frac{\nu_{i+1}-r_{i,1}}{\nu_{i+1}-r_{i,2}}\right) - \ln\left(\frac{\nu_i-r_{i,1}}{\nu_i-r_{i,2}}\right)}{r_{i,1}-r_{i,2}} & \text{si } \Delta_i > 0 \text{ avec } r_{i,j} = \frac{-b_i+(-1)^j\sqrt{\Delta_i}}{2a_i}. \end{cases}$$

Si f est C^3 sur $[a, b]$, on a le résultat classique d'interpolation [29] :

$$\forall x \in [a, b], |f(x) - f_h(x)| = O(h^3).$$

Grâce à l'inégalité fondamentale, on conclut que $x_h(t)$ approche $x(t)$ avec une précision d'ordre $O(h^3)$.

Expérimentalement, on trouve que l'approximation est d'ordre $O(h^4)$. Ceci peut s'expliquer simplement, en reprenant l'évaluation de l'erreur donnée par (10.3.5). Il est facile de montrer, grâce à la théorie de l'intégration numérique [29], que si f est C^4 , alors pour tout $i \in \{i_0 + 1, \dots, i_t - 1\}$,

$$\int_{\nu_i}^{\nu_{i+1}} \left(\frac{1}{f_h(x)} - \frac{1}{f(x)} \right) = O(h^5).$$

On a alors,

$$|x(t) - x_h(t)| \leq \sup_{x \in [a, b]} |f(x)| \left[O(h^4) + \sum_{i=i_0+1}^{i=i_t-1} O(h^5) \right] = O(h^4).$$

Proposition 10.3.2 *Soit x la solution de l'équation différentielle (10.2.1), où le champ de vecteurs f est C^4 sur $[a, b]$; soit x_h la solution de l'équation différentielle (10.2.2), où le champ de vecteurs f_h est défini par (10.3.6).*

Pour tout t , où $x(t)$ et $x_h(t)$ sont définies,

$$|x(t) - x_h(t)| = O(h^4).$$

Nous avons donc proposé deux méthodes permettant d'augmenter l'ordre de convergence. On peut, bien sûr, utiliser une méthode hybride (interpolation quadratique par morceaux aux abscisses de Gauss [45]), ou encore augmenter l'ordre d'approximation (splines cubiques).

10.4 Résultats expérimentaux

Dans cette partie, nous comparons les performances des différentes méthodes présentées ici avec celles des méthodes classiques de Runge-Kutta (voir par exemple [29, 32]). Dans les tableaux suivants, ces méthodes sont répertoriées par les sigles :

- **RKN** : Méthode de Runge-Kutta d'ordre N .
- **PLI** : Méthode de calcul hybride utilisant l'interpolant linéaire par morceaux aux sommets de la discrétisation.
- **GPLI** : Méthode de calcul hybride utilisant l'interpolant linéaire par morceaux aux abscisses de Gauss de la discrétisation.
- **PQI** : Méthode de calcul hybride utilisant l'interpolant quadratique par morceaux.

Ces différentes méthodes ont été implémentées² en C++. Pour notre expérimentation numérique, nous considérons l'équation différentielle

$$x'(t) = e^{x(t)}, \quad x(0) = 1$$

dont la solution est

$$x(t) = \ln \left(\frac{1}{e^{-1} - t} \right).$$

Pour estimer la qualité d'approximation des différentes méthodes, nous avons évalué l'erreur à l'instant $t = 0.3$. Les résultats sont répertoriés dans les tableaux 10.2 et 10.3.

²en collaboration avec Jean-Guillaume Dumas

Nombre d'itérations	PLI	GPLI	RK3	PQI	RK4
10	1.48e-2	6.13e-5	1.45e-3	2.13e-5	1.40e-5
20	3.69e-3	3.42e-6	2.07e-4	1.10e-6	1.55e-6
40	9.22e-4	8.32e-7	2.76e-5	7.38e-8	1.23e-7
80	2.30e-4	5.22e-8	3.56e-6	5.13e-9	8.66e-9
160	5.75e-5	1.63e-8	4.52e-7	2.88e-10	5.71e-10
320	1.43e-5	9.87e-10	5.70e-8	2.00e-11	3.66e-11
640	3.59e-6	2.40e-10	7.15e-9	1.11e-12	2.32e-12
1280	8.99e-7	1.70e-11	8.96e-10	7.95e-14	1.46e-13

TAB. 10.2: Résultats expérimentaux pour $x' = e^x$, $x(0) = 1$. Erreur à $t = 0.3$.

Méthode	Temps moyen de calcul pour 1280 itérations (ms)	Ordre expérimental	Ordre théorique
PLI	1.124	2.001	2
GPLI	1.168	3.058	3
RK3	0.677	2.968	3
PQI	1.597	4.210	4
RK4	0.843	3.897	4

TAB. 10.3: Résultats expérimentaux pour $x' = e^x$, $x(0) = 1$. (Pentium 3, 1 GHz, C++).

On voit donc que l'ordre expérimental des méthodes présentées précédemment est conforme à celui que prédisait la théorie.

Nous avons, également, calculé le temps de calcul de ces méthodes (moyenne sur 1000 exécutions). On peut voir que les méthodes de calcul hybride sont légèrement plus chères que les méthodes de Runge-Kutta, cependant le coût reste comparable.

Ainsi, les bons résultats du calcul hybride sur les équations différentielles scalaires nous encourage à pousser plus loin notre étude. Dans le chapitre suivant, nous généralisons la méthode d'interpolation linéaire par morceaux aux sommets d'une subdivision de l'espace des phases à des équations différentielles de plus grandes dimensions.

Chapitre 11

Calcul Hybride en Dimension Supérieure

Dans ce chapitre, nous généralisons l'approche, décrite dans le chapitre précédent, aux équations différentielles ordinaires de dimension supérieure ou égale à 2. On considère donc une équation différentielle :

$$x'(t) = f(x(t)), \quad x(t) \in D, \quad (11.0.1)$$

où D est un compact de \mathbb{R}^n .

L'idée de la méthode reste identique, on définit une partition du domaine D . Sur chaque élément de la partition, le champ de vecteurs f est approché localement par un champ de vecteurs plus simple. On résout, ensuite, l'équation différentielle par morceaux ainsi définie.

Notons que, contrairement au cas scalaire, le choix de l'approximation de f est plus limité. En effet, si en dimension 1, les équations différentielles, associées à un champ de vecteurs quadratique, sont bien connues, ce n'est pas le cas en dimension supérieure. Ainsi, le choix d'une approximation linéaire par morceaux s'impose. De même, il convient de remarquer que l'estimation de l'erreur d'approximation donnée par l'inégalité (10.3.3) n'est plus valable. En effet, le changement de variable, menant à ce résultat, n'est possible que pour des équations différentielles scalaires.

Nous allons donc nous intéresser à la généralisation de la méthode d'approximation du champ de vecteurs par interpolation aux sommets des éléments de la partition. Certains des résultats exposés dans ce chapitre sont issus de [6] et [46].

11.1 Approximation de l'équation différentielle

Nous voulons approcher l'équation différentielle (11.0.1) par :

$$x'_h(t) = f_h(x_h(t)), \quad x_h(t) \in D, \quad (11.1.2)$$

où f_h est une approximation linéaire par morceaux du champ de vecteurs f .

11.1.1 Approximation linéaire par morceaux d'un champ de vecteurs

Soit donc $(D_i)_{i \in I}$, une partition du domaine D , où chaque élément D_i est un polytope.

Calcul de l'approximation

Nous voulons calculer, par interpolation aux sommets de D_i , une approximation linéaire f_i du champ de vecteurs f :

$$f_i(x) = A_i x + b_i.$$

En dimension n , un champ de vecteurs linéaire f_i , est caractérisé, de manière unique, par sa valeur en $n + 1$ points, $\{y_0, \dots, y_n\}$, affinement indépendants :

$$\det(y_1 - y_0, \dots, y_n - y_0) \neq 0.$$

On suppose donc que, pour tout $i \in I$, le polytope D_i est l'enveloppe convexe de $n + 1$ points affinement indépendants, $\{y_0^i, \dots, y_n^i\}$. Ainsi [18], pour tout $i \in I$, D_i est un simplexe de \mathbb{R}^n .

L'approximation de f , sur le domaine D_i , est alors donnée par les contraintes d'interpolation aux sommets du simplexe :

$$\forall j \in \{0, \dots, n\}, A_i y_j^i + b_i = f(y_j^i).$$

Donc,

$$\forall j \in \{1, \dots, n\}, A_i (y_j^i - y_0^i) = f(y_j^i) - f(y_0^i).$$

On note Y_i (respectivement F_i) la matrice $n \times n$ dont les colonnes sont les vecteurs $\{y_j^i - y_0^i, j \in \{1, \dots, n\}\}$ (respectivement $\{f(y_j^i) - f(y_0^i), j \in \{1, \dots, n\}\}$).

Les contraintes d'interpolation, sous forme matricielle, deviennent $A_i Y_i = F_i$. Par indépendance affine des sommets du simplexe, la matrice Y_i est inversible. L'approximation linéaire du champ de vecteurs f sur D_i est donc donnée par :

$$A_i = F_i Y_i^{-1}, \text{ et } b_i = f(y_0^i) - A_i y_0^i.$$

L'approximation linéaire par morceaux de f est définie par

$$f_h(x) = f_i(x), \text{ si } x \in D_i. \quad (11.1.3)$$

L'équation différentielle approchée (11.1.2) définit ainsi, clairement, un système hybride linéaire par morceaux.

Convergence de l'approximation

Nous avons vu, dans le chapitre précédent (voir théorème 10.2.1), que les solutions de l'équation différentielle (11.1.2) approchaient les solutions de (11.0.1) avec une précision du même ordre que l'approximation du champ de vecteurs f par f_h . Il est donc important d'évaluer la qualité de cette approximation.

Soit $i \in I$, on note h_i la taille du simplexe D_i :

$$h_i = \sup_{x, y \in D_i} \|x - y\|.$$

où $\|\cdot\|$ dénote la norme ∞ sur \mathbb{R}^n . La taille de la partition $(D_i)_{i \in I}$ est $h = \sup_{i \in I} h_i$.

On peut, sous certaines hypothèses sur le champ de vecteurs f , majorer l'erreur d'interpolation en fonction de la taille de la partition h .

Proposition 11.1.1 (Erreur d'interpolation) *Si f est L -Lipschitz sur D ,*

$$\sup_{x \in D} \|f(x) - f_h(x)\| \leq \frac{2Ln}{n+1}h.$$

Preuve : Soit $x \in D$, il existe $i \in I$, tel que $x \in D_i$. Soit $\{y_0^i, \dots, y_n^i\}$, l'ensemble des sommets du simplexe D_i . Il existe $n+1$ scalaires positifs $\alpha_0, \dots, \alpha_n$, tels que

$$x = \sum_{k=0}^{k=n} \alpha_k y_k^i, \quad \sum_{k=0}^{k=n} \alpha_k = 1.$$

Notons qu'il existe $l \in \{0, \dots, n\}$, tel que $\alpha_l \geq \frac{1}{n+1}$. Par l'inégalité triangulaire,

$$\|f(x) - f_h(x)\| = \|f(x) - f_i(x)\| \leq \|f(x) - f_i(y_l^i)\| + \|f_i(y_l^i) - f_i(x)\|.$$

Les contraintes d'interpolation donnent $f(y_l^i) = f_i(y_l^i)$. De plus, f est L -Lipschitz, donc

$$\|f(x) - f_i(y_l^i)\| \leq \|f(x) - f(y_l^i)\| \leq L\|x - y_l^i\|.$$

En reportant l'expression barycentrique de x :

$$\|f(x) - f_i(y_l^i)\| \leq L \left\| \sum_{k=0}^{k=n} \alpha_k y_k^i - y_l^i \right\| \leq L \sum_{k=0}^{k=n} \alpha_k \|y_k^i - y_l^i\|.$$

Pour tout $k \neq l$, $\|y_k^i - y_l^i\| \leq h$, on a donc

$$\|f(x) - f_i(y_l^i)\| \leq L(1 - \alpha_l)h.$$

Par ailleurs, par linéarité du champ de vecteurs f_i ,

$$\|f_i(y_l^i) - f_i(x)\| = \|f_i(y_l^i) - \sum_{k=0}^{k=n} \alpha_k f_i(y_k^i)\| \leq \sum_{k=0}^{k=n} \alpha_k \|f_i(y_l^i) - f_i(y_k^i)\|.$$

En remarquant que pour tout $k \in \{0, \dots, n\}$, $f_i(y_k^i) = f(y_k^i)$ et comme f est L -Lipschitz :

$$\|f_i(y_l^i) - f_i(x)\| \leq \sum_{k=0}^{k=n} \alpha_k \|f(y_l^i) - f(y_k^i)\| \leq L \sum_{k=0}^{k=n} \alpha_k \|y_l^i - y_k^i\| \leq L(1 - \alpha_l)h.$$

On a donc, finalement,

$$\|f(x) - f_h(x)\| \leq 2L(1 - \alpha_l)h. \leq 2L \frac{n}{n+1}h.$$

■

Si le champ de vecteurs f est Lipschitz sur D , alors l'approximation de f par f_h est d'ordre $O(h)$, et donc linéaire dans la taille de la partition.

En faisant des hypothèses supplémentaires sur la régularité de f , on peut montrer que l'approximation est d'ordre supérieur.

Proposition 11.1.2 (Erreur d'interpolation) *Si f est C^2 sur D ,*

$$\sup_{x \in D} \|f(x) - f_h(x)\| \leq \frac{Mn^2}{2(n+1)^2} h^2$$

avec

$$M = \max_{j=1}^{j=n} \sup_{x \in D} \sum_{p_1=1}^{p_1=n} \sum_{p_2=1}^{p_2=n} \left| \frac{\partial^2 f^j(x)}{\partial x_{p_1} \partial x_{p_2}} \right|$$

où f^j dénote la j -ième composante de f .

Preuve : Soit $x \in D$, il existe $i \in I$, tel que $x \in D_i$. Soit $\{y_0^i, \dots, y_n^i\}$ l'ensemble des sommets du simplexe D_i . On définit la fonction erreur

$$e_h(x) = f(x) - f_h(x) = f(x) - f_i(x).$$

Soit $j \in \{1, \dots, n\}$, $e_h^j(x)$ dénote la j -ième composante du vecteur $e_h(x)$.

D_i est compact, et la fonction $e_h^j(x)$ est continue sur D_i donc

$$\bar{x}^j = \arg \max_{x \in D_i} |e_h^j(x)| \in D_i.$$

Il existe donc $n+1$ scalaires positifs, $\alpha_0, \dots, \alpha_n$, tels que

$$\bar{x}^j = \sum_{k=0}^{k=n} \alpha_k y_k^i, \quad \sum_{k=0}^{k=n} \alpha_k = 1.$$

Nécessairement, il existe $l \in \{0, \dots, n\}$, tel que $\alpha_l \geq \frac{1}{n+1}$.

Si $\bar{x}^j = y_l^i$, alors pour tout $x \in D_i$, $|e_h^j(x)| \leq |e_h^j(y_l^i)| = 0$.

Supposons $\bar{x}^j \neq y_l^i$ (donc $\alpha_l \neq 1$). Les points de la droite de \mathbb{R}^n , passant par \bar{x}^j et y_l^i , peuvent être paramétrés de la manière suivante

$$x(\lambda) = \bar{x}^j + \lambda(y_l^i - \bar{x}^j) = (\alpha_l + \lambda(1 - \alpha_l))y_l^i + \sum_{k=0, k \neq l}^{k=n} (1 - \lambda)\alpha_k y_k^i.$$

Remarquons que, pour tout $\lambda \in [\frac{-\alpha_l}{1-\alpha_l}, 1]$,

$$\alpha_l + \lambda(1 - \alpha_l) \geq 0 \text{ et } \forall k \neq l, (1 - \lambda)\alpha_k \geq 0.$$

Donc, pour tout $\lambda \in [\frac{-\alpha_l}{1-\alpha_l}, 1]$, $x(\lambda) \in D_i$.

On définit alors, sur l'intervalle $[\frac{-\alpha_l}{1-\alpha_l}, 1]$, l'application ϵ

$$\epsilon(\lambda) = e_h^j(x(\lambda)) = f^j(x(\lambda)) - f_i^j(x(\lambda)).$$

ϵ est C^2 sur $[\frac{-\alpha_l}{1-\alpha_l}, 1]$, et

$$\epsilon''(\lambda) = \sum_{p_1=1}^{p_1=n} \sum_{p_2=1}^{p_2=n} \frac{\partial^2 e_h^j}{\partial x_{p_1} \partial x_{p_2}}(x(\lambda))(y_{l,p_1}^i - \bar{x}_{p_1}^j)(y_{l,p_2}^i - \bar{x}_{p_2}^j).$$

où $y_{l,p}^i$ et \bar{x}_p^j sont les p -ièmes composantes des vecteurs y_l^i et \bar{x}^j .
Or, pour tout $p \in \{1, \dots, n\}$,

$$|y_{l,p}^i - \bar{x}_p^j| \leq \|y_l^i - \bar{x}^j\| \leq \sum_{k=0}^{k=n} \alpha_k \|y_l^i - y_k^i\| \leq (1 - \alpha_l) h_i \leq \frac{n}{n+1} h.$$

De plus, pour tout $x \in D_i$,

$$\frac{\partial^2 e_h^j}{\partial x_{p_1} \partial x_{p_2}}(x) = \frac{\partial^2 f^j}{\partial x_{p_1} \partial x_{p_2}}(x) - \frac{\partial^2 f_h^j}{\partial x_{p_1} \partial x_{p_2}}(x) = \frac{\partial^2 f^j}{\partial x_{p_1} \partial x_{p_2}}(x)$$

D'où, pour tout $\lambda \in [\frac{-\alpha_l}{1-\alpha_l}, 1]$,

$$|\epsilon''(\lambda)| \leq \sum_{p_1=1}^{p_1=n} \sum_{p_2=1}^{p_2=n} \left| \frac{\partial^2 f^j}{\partial x_{p_1} \partial x_{p_2}}(x(\lambda)) \right| \frac{n^2}{(n+1)^2} h^2 \leq M \frac{n^2}{(n+1)^2} h^2.$$

Grâce à la formule de Taylor avec reste intégral

$$\epsilon(1) = \epsilon(0) + \epsilon'(0) + \int_0^1 \epsilon''(\lambda)(1-s) ds.$$

$x(1) = y_l^i$, donc $\epsilon(1) = 0$. $x(0) = \bar{x}^j$, donc $\epsilon(\lambda)$ atteint un extremum en $\lambda = 0$. Par conséquent, $\epsilon'(0) = 0$. D'où,

$$\epsilon(0) = - \int_0^1 \epsilon''(\lambda)(1-s) ds.$$

On a donc, pour tout $x \in D$,

$$|f^j(x) - f_h^j(x)| = |e_h^j(x)| \leq \epsilon(0) \leq M \frac{n^2}{(n+1)^2} h^2 \int_0^1 (1-s) ds.$$

Donc,

$$|f^j(x) - f_h^j(x)| \leq \frac{Mn^2}{2(n+1)^2} h^2.$$

■

Ainsi, si le champ de vecteurs f est deux fois différentiable sur D , l'ordre de l'approximation par f_h est $O(h^2)$.

Continuité de l'approximation

Nous avons vu que pour pouvoir définir f_h comme le champ de vecteurs linéaire par morceaux, interpolant f aux sommets des polytopes de la partition $(D_i)_{i \in I}$, il fallait que les éléments de la partition D_i soient des simplexes.

En imposant des contraintes supplémentaires, on peut garantir que le champ de vecteurs f_h est continu et Lipschitz sur D . Cette propriété se révèle intéressante, puisque dans ce cas, l'équation différentielle approchée (11.1.2) admet une et une seule solution pour une condition initiale donnée.

Définition 11.1.1 (Maillage) Une famille de simplexes $(D_i)_{i \in I}$ est un maillage (voir figure 11.1) d'un polytope D si

1. $\bigcup_{i \in I} D_i = D$.
2. $\forall i_1, i_2 \in I, i_1 \neq i_2$, l'intersection $D_{i_1} \cap D_{i_2}$ est soit vide, soit égale à l'enveloppe convexe des sommets de D_{i_1} communs à D_{i_2} .

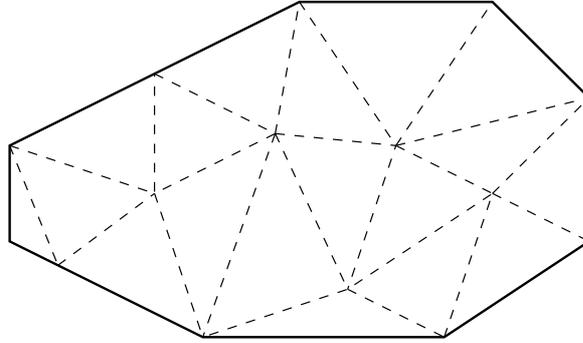


FIG. 11.1: Exemple de maillage d'un polytope

Proposition 11.1.3 Si $(D_i)_{i \in I}$ est un maillage de D , le champ de vecteurs f_h est continu et L_h -Lipschitz sur D avec

$$L_h = \max_{i \in I} \|A_i\|.$$

Preuve : Tout d'abord, il est clair que f_h est continu sur l'intérieur de chaque simplexe D_i . Il suffit donc, de vérifier que le champ de vecteurs se recolle continûment à l'intersection de deux simplexes.

Soient donc D_{i_1} et D_{i_2} deux simplexes, distincts, de la subdivision tels que $D_{i_1} \cap D_{i_2} \neq \emptyset$. Notons $\{y_0, \dots, y_p\}$ leur sommets communs ($p \in \{0, \dots, n-1\}$).

Soit x un point de $D_{i_1} \cap D_{i_2}$, il existe p scalaires positifs tels que

$$x = \sum_{k=0}^{k=p} \alpha_k y_k.$$

Par linéarité de f_{i_1} et f_{i_2} :

$$f_{i_1}(x) = \sum_{k=0}^{k=p} \alpha_k f_{i_1}(y_k) = \sum_{k=0}^{k=p} \alpha_k f(y_k) = \sum_{k=0}^{k=p} \alpha_k f_{i_2}(y_k) = f_{i_2}(x).$$

Par conséquent, f_h est continu sur le polytope D .

Soient $x \in D$, $y \in D$; si x et y appartiennent au même simplexe du maillage D_i . On a alors

$$\|f_h(x) - f_h(y)\| = \|A_i(x - y)\| \leq \|A_i\| \|x - y\|.$$

Sinon, le segment qui relie x à y traverse une séquence de $p + 1$ simplexes du maillage, $D_{i_1}, \dots, D_{i_{p+1}}$. On a $x \in D_{i_1}$ et $y \in D_{i_{p+1}}$. Notons x_k l'intersection du segment qui relie x à y avec $D_{i_k} \cap D_{i_{k+1}}$, $x_0 = x$ et $x_{p+1} = y$.

$$\begin{aligned} \|f_h(x) - f_h(y)\| &\leq \sum_{k=0}^{k=p} \|f_h(x_k) - f_h(x_{k+1})\| \leq \sum_{k=0}^{k=p} \|A_{i_{k+1}}(x_k - x_{k+1})\| \\ &\leq \sum_{k=0}^{k=p} \|A_{i_{k+1}}\| \|x_k - x_{k+1}\| \leq L_h \sum_{k=0}^{k=p} \|x_k - x_{k+1}\|. \end{aligned}$$

$x_k - x_{k+1}$ est colinéaire à $x - y$, on a donc $x_k - x_{k+1} = \lambda_k(x - y)$ avec $\lambda_k \geq 0$. Remarquons que

$$x - y = \sum_{k=0}^{k=p} x_k - x_{k+1} = \sum_{k=0}^{k=p} \lambda_k(x - y).$$

Donc, $\sum_{k=0}^{k=p} \lambda_k = 1$. D'où

$$\|f_h(x) - f_h(y)\| \leq L_h \sum_{k=0}^{k=p} \|x_k - x_{k+1}\| = L_h \sum_{k=0}^{k=p} \lambda_k \|x - y\| = L_h \|x - y\|.$$

■

Si la famille de simplexes $(D_i)_{i \in I}$ est un maillage de D , alors, pour toute condition initiale $x_0 \in D$, il existe une unique solution maximale au problème de Cauchy :

$$x'_h(t) = f_h(x_h(t)), \quad x_h(t) \in D, \quad x_h(0) = x_0.$$

De plus, cette solution est continûment dérivable.

11.1.2 Maillage simplicial du domaine

Nous avons vu, comment, à partir d'un maillage du domaine D , on pouvait construire une approximation f_h du champ de vecteurs f , continue, Lipschitz sur D et de bonne qualité.

Cependant, nous n'avons pas encore parlé de la construction d'un maillage du domaine D . On peut bien sûr avoir recours à une méthode de triangulation de Delaunay en dimension n [53]. Cette méthode possède le désavantage de devoir précalculer toute la triangulation et de la stocker en mémoire. Lorsque la dimension augmente, l'espace mémoire requis peut alors devenir prohibitif.

Une solution alternative, est de définir implicitement un maillage du domaine D . Ainsi, la triangulation se fait, à la volée et localement.

Supposons que le domaine D est un pavé de \mathbb{R}^n , $D = [a_1, b_1] \times \cdots \times [a_n, b_n]$. On découpe, de manière implicite, le domaine D en cubes $(C_j)_{j \in J}$ dont les arêtes sont de longueur h . Pour la résolution de l'équation différentielle (11.1.2), pour une condition initiale x_0 , il est facile de calculer le cube C_j contenant x_0 . On construit alors un maillage de C_j . On intègre l'équation différentielle, jusqu'à ce que la solution sorte de C_j pour rentrer dans un autre cube $C_{j'}$. On recommence ensuite en construisant un maillage de $C_{j'}$.

Il nous faut donc détailler comment on peut construire un maillage d'un cube de \mathbb{R}^n . On doit aussi garantir que l'union des maillages de C_j forment un maillage du domaine D .

Maillage d'un cube de \mathbb{R}^n

Considérons le cube unité de \mathbb{R}^n , $C = [0, 1]^n$. Soit Π , l'ensemble des permutations de $\{1, \dots, n\}$. Π possède $n!$ éléments distincts. Soit $\pi \in \Pi$, $\pi = (k_1, \dots, k_n)$, on définit l'ensemble S_π (voir figure 11.2) de la manière suivante.

$$S_\pi = \{x \in \mathbb{R}^n, 0 \leq x_{k_1} \leq \cdots \leq x_{k_n} \leq 1\}.$$

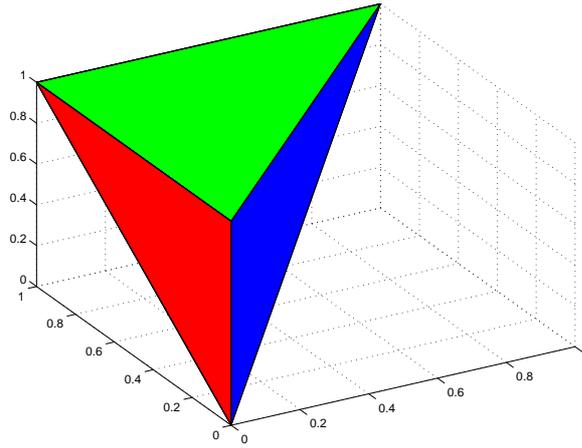


FIG. 11.2: Exemple de simplexe S_π en dimension 3.

S_π est donc un simplexe. Les $(n + 1)$ sommets de S_π sont les points dont les composantes vérifient :

$$\begin{cases} \forall i \in \{1, \dots, p\}, & x_{k_i} = 0 \\ \forall i \in \{p + 1, \dots, n\} & x_{k_i} = 1 \end{cases}, p \in \{0, \dots, n\}.$$

Lemme 11.1.1 (Maillage d'un n -cube [42]) *La partition $(S_\pi)_{\pi \in \Pi}$ forme un maillage de $[0, 1]^n$.*

Utilisons donc ce résultat pour construire un maillage du pavé D .

Maillage de D

Supposons que le domaine D soit un pavé de la forme :

$$D = [a_1, a_1 + (N_1 + 1)h] \times \cdots \times [a_n, a_n + (N_n + 1)h].$$

Le domaine D peut donc être partitionné en cubes dont les arêtes sont de longueur h :

$$C_{i_1, \dots, i_n} = [a_1 + i_1 h, a_1 + (i_1 + 1)h] \times \dots \times [a_n + i_n h, a_n + (i_n + 1)h]$$

$$i_1 \in \{0, \dots, N_1\}, \dots, i_n \in \{0, \dots, N_n\}.$$

Chaque cube de la partition est, à son tour, subdivisé en $n!$ simplexes par la méthode décrite précédemment. Le partition résultante est un maillage de D (voir [66]). De plus, il est clair que la taille du maillage ainsi généré est égale à h .

11.2 Approximation des solutions de l'équation différentielle

Nous montrons dans cette partie les propriétés de convergence des solutions de l'équation différentielle approchée vers les solutions de l'équation initiale.

11.2.1 Approximation de la solution du problème de Cauchy

Soit x_0 un point du domaine de l'équation différentielle. On note $x(t)$, la solution de l'équation différentielle (11.0.1), pour la condition initiale $x(0) = x_0$. $x_h(t)$ est la solution de l'équation différentielle approchée (11.1.2), pour la condition initiale $x(0) = x_0$, où le champ de vecteurs f_h est calculé par la méthode présentée précédemment.

On note $\varepsilon(h)$ l'erreur d'approximation de f par f_h donnée par les lemmes 11.1.1 ou 11.1.2.

Théorème 11.2.1 (Convergence C^1 de l'approximation) *Pour tout t , où $x(t)$ et $x_h(t)$ sont définies,*

$$\|x(t) - x_h(t)\| \leq \frac{\varepsilon(h)}{L} (e^{L|t|} - 1)$$

et

$$\|x'(t) - x'_h(t)\| \leq \varepsilon(h) e^{L|t|}.$$

Preuve : D'après l'inégalité fondamentale (théorème 10.2.1), on a de manière immédiate la première inégalité. Montrons la deuxième inégalité,

$$\begin{aligned} \|x'(t) - x'_h(t)\| &= \|f(x(t)) - f_h(x_h(t))\| \\ &\leq \|f(x(t)) - f(x_h(t))\| + \|f(x_h(t)) - f_h(x_h(t))\| \\ &\leq L\|x(t) - x_h(t)\| + \varepsilon(h) \\ &\leq \varepsilon(h) e^{L|t|}. \end{aligned}$$

■

Exemple 11.2.1 *On considère l'équation différentielle de dimension 2*

$$\begin{cases} x'_1(t) &= -x_1(t) + x_2(t) + (x - 1.85)^2 \\ x'_2(t) &= 3x_1(t) + x_2(t) - x_1^3(t). \end{cases} \quad (11.2.4)$$

Les approximations des solutions de l'équation différentielle sont calculées, en appliquant l'algorithme de simulation, présenté dans le chapitre 4, au système hybride linéaire par morceaux défini par le champ de vecteurs f_h .

Un exemple d'approximation de solution, tracée dans l'espace des phases (x_1, x_2) , est représenté sur la figure 11.3, les éléments du maillage du domaine, traversés par l'approximation, sont également tracés.

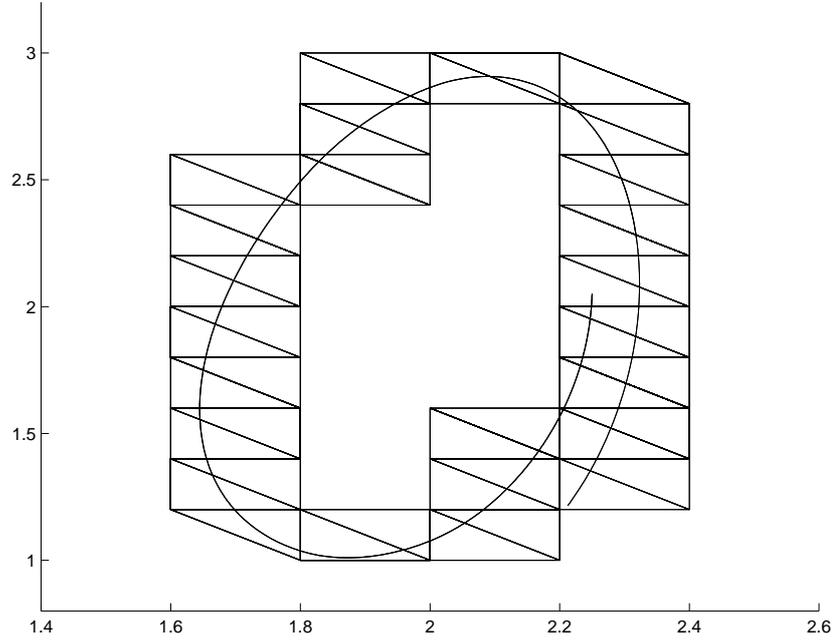


FIG. 11.3: Solution approchée de l'équation différentielle (11.2.4) ainsi que le maillage associé.

11.2.2 Approximation des attracteurs

Le théorème 11.2.1 montre donc que, pour tout t , $x_h(t)$ tend vers $x(t)$. Cependant, il convient de remarquer que la majoration de $\|x_h(t) - x(t)\|$ croît exponentiellement par rapport à t .

On peut donc légitimement se demander si le résultat de convergence reste vrai lorsque t tend vers l'infini. Notamment, si lorsque t tend vers l'infini, $x(t)$ tend vers un attracteur de l'équation différentielle (11.0.1), il est intéressant de savoir si $x_h(t)$ tend également vers un attracteur de l'équation différentielle (11.1.2).

Ensuite, l'attracteur de $x_h(t)$ tend-il vers celui de $x(t)$ lorsque h tend vers 0?

Les résultats exposés dans cette partie sont issus de [89] et sont des conséquences du théorème 11.2.1.

Notons Φ le flot de l'équation différentielle (11.0.1).

Définition 11.2.1 (Attracteur) *Un attracteur est un sous-ensemble de D , compact et invariant par le flot Φ :*

$$\forall t \geq 0, \Phi(A, t) \subseteq A,$$

et tel qu'il existe B , un voisinage ouvert de A vérifiant :

$$\forall x \in B, \lim_{t \rightarrow +\infty} \left(\inf_{y \in A} \|\Phi(x, t) - y\| \right) = 0$$

Supposons que l'équation différentielle (11.0.1) a un attracteur A , qui attire un voisinage B invariant par le flot Φ . Alors, on peut montrer le résultat suivant.

Théorème 11.2.2 (Existence et semi-continuité supérieure des attracteurs) *Il existe $h_0 > 0$, tel que pour tout $0 < h < h_0$, l'équation différentielle approchée (11.1.2) possède un attracteur $A_h \subseteq B$ et qui attire B .*

De plus,

$$\lim_{h \rightarrow 0} \left(\sup_{x \in A_h} \inf_{y \in A} \|x - y\| = 0 \right).$$

Ainsi, on peut voir que l'existence d'un attracteur pour l'équation différentielle (11.0.1) implique l'existence d'un attracteur pour l'équation différentielle approchée (11.1.2). On n'a pas, en général, la convergence de l'attracteur A_h vers l'attracteur A . A_h est inclus dans un voisinage de A . La réciproque est généralement fautive, cependant, il existe certains cas où l'on peut montrer la convergence (voir par exemple [89]).

Exemple 11.2.2 *Reprenons l'équation différentielle (11.2.4), cette équation possède un cycle limite. Donc, pour h suffisamment petit, l'équation différentielle approchée possède également un attracteur.*

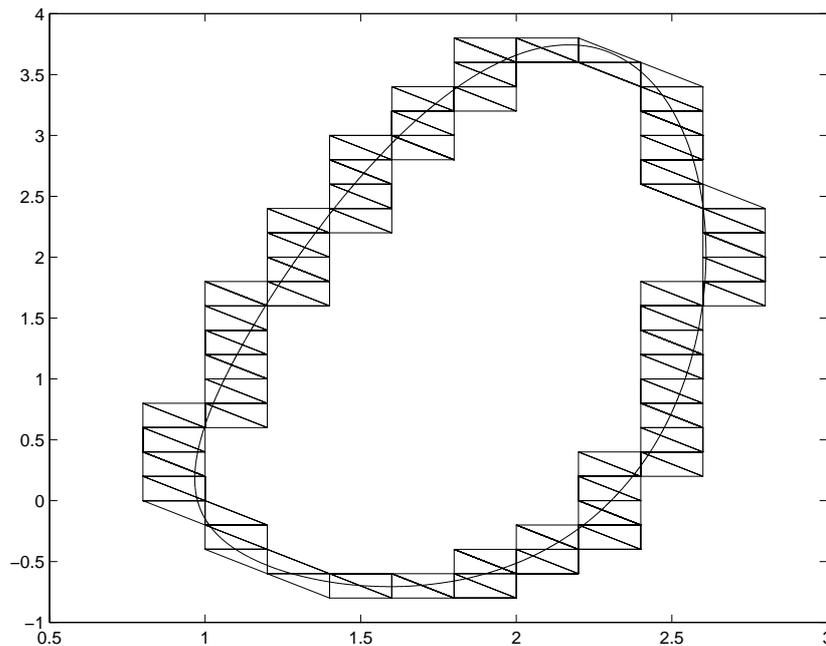


FIG. 11.4: Approximation du cycle limite de l'équation différentielle (11.2.4), ainsi que le maillage associé.

Quelques simulations rapides montrent qu'il s'agit également d'un cycle limite. Un calcul précis et efficace de ce cycle peut être effectué grâce à la méthode de calcul des exécutions périodiques présentée dans le chapitre 5.

Cette solution nous donne donc une approximation du cycle limite de l'équation différentielle (11.2.4). Le résultat est dessiné sur la figure 11.4.

11.2.3 Comparaison des portraits de phases

Après les questions de la convergence des solutions et des attracteurs. Il est intéressant d'examiner celle de la convergence du portrait de phase.

Plutôt que de parler de convergence du portrait de phase, il est plus judicieux de considérer la notion d'équivalence topologique de deux champs de vecteurs.

Définition 11.2.2 (Equivalence topologique) Soient f_1 et f_2 , deux champs de vecteurs définis sur des régions U_1 et U_2 de \mathbb{R}^n . On dit que f_1 est topologiquement équivalent à f_2 , si il existe un homéomorphisme de U_1 dans U_2 qui transforme les trajectoires orientées de l'équation différentielle $x' = f_1(x)$ en les trajectoires orientées de l'équation différentielle $x' = f_2(x)$.

Globalement, cela signifie qu'à un point d'équilibre de l'équation différentielle $x' = f_1(x)$ correspond un point d'équilibre de $x' = f_2(x)$, avec les mêmes propriétés; un point stable est transformé en un point stable, un point instable en un point instable, un point selle en un point selle. De même, une solution périodique de $x' = f_1(x)$ correspond à une solution périodique de $x' = f_2(x)$ avec les mêmes propriétés de stabilité.

Ainsi, la question de la convergence du portrait de phase est de savoir si, il existe un $h_0 > 0$, tel que pour tout $h \leq h_0$, le champ de vecteurs f et son approximation linéaire par morceaux f_h sont topologiquement équivalents.

Un champ de vecteurs f topologiquement équivalent à toute petite perturbation de lui même est appelé structurellement stable. Si f est structurellement stable, il apparaît donc clairement que, pour h suffisamment petit, f et f_h sont topologiquement équivalents.

Pour des champs de vecteurs plans, des conditions existent pour garantir qu'un champ de vecteurs est structurellement stable (voir par exemple [62]). Malheureusement, il est généralement impossible de vérifier, simplement, si ces conditions sont remplies.

Exemple 11.2.3 Reprenons l'équation différentielle scalaire, étudiée dans le chapitre précédent,

$$x'(t) = 1 + \sin(x(t)).$$

Cette équation différentielle possède un point d'équilibre $x = 3\pi/2$.
Considérons maintenant la légère perturbation de cette équation

$$x'(t) = 1 + \sin(x(t)) + \varepsilon$$

où ε est un réel strictement positif, aussi petit que l'on veut.

Pour tout ε , l'équation différentielle perturbée ne possède pas de point d'équilibre. On peut donc conclure que le champ de vecteurs n'est pas structurellement stable.

Il est donc normal que les méthodes de calcul hybride échouent à capturer le comportement asymptotique des solutions.

Exemple 11.2.4 Reprenons l'équation différentielle (11.2.4).

Nous avons calculé, par deux méthodes différentes, les portraits de phase sur le domaine $[-5, 3] \times [-5, 5]$.

Sur la figure 11.5, nous avons tracé le portrait de phase calculé par le logiciel Matlab en utilisant une méthode de Runge-Kutta ainsi que le portrait de phase calculé par une

méthode de calcul hybride, à l'aide de l'outil CASCADE (voir [51] ou en annexe).

On peut constater que les champs de vecteurs sont topologiquement équivalents. Il y a deux points d'équilibre, un instable et un point selle, ainsi qu'un cycle limite stable.

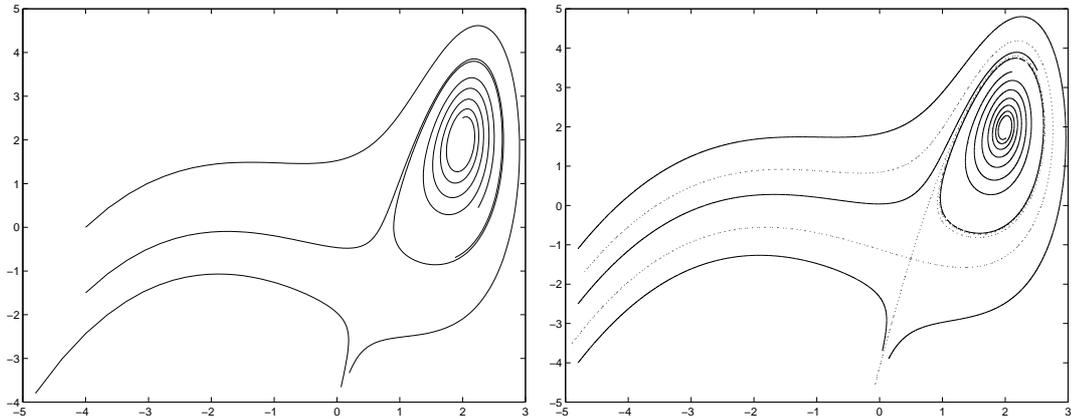


FIG. 11.5: Portrait de phase de l'équation différentielle (11.2.4), calculé par une méthode de Runge-Kutta (à gauche), et par une méthode de calcul hybride (à droite).

11.3 Approximation conservative

On voit donc que, souvent, l'étude de l'équation différentielle approchée nous permet d'avoir de bonnes informations, aussi bien qualitatives que quantitatives sur l'équation différentielle originale. Malheureusement, il n'est généralement pas possible de préciser si l'approximation est correcte ou non.

Ceci peut se révéler problématique, à partir du moment où l'on ne s'intéresse pas seulement au comportement des solutions de l'équation différentielle à court terme, mais aussi à long terme. Typiquement, certaines applications telles que la vérification nécessitent le calcul de l'ensemble atteignable par les solutions de l'équation différentielle (11.0.1) avec des valeurs initiales dans un ensemble I :

$$\mathcal{R}(I) = \bigcup_{y \in I, s \in \mathbb{R}^+} \Phi(y, s)$$

où Φ dénote le flot de l'équation différentielle (11.0.1).

Une idée, pour approcher cet ensemble, est de calculer l'ensemble atteignable par les solutions de l'équation différentielle approchée (11.1.2).

Exemple 11.3.1 Reprenons l'exemple $x' = 1 + \sin(x)$. Pour un ensemble de valeurs initiales réduit au singleton $\{1\}$, l'ensemble atteignable est clairement $[1, 3\pi/2]$.

Or, pour tout $\varepsilon > 0$, il est clair qu'il existe $h \leq \varepsilon$ tel que l'ensemble atteignable par les solutions de l'équation approchée soit égal à $[1, +\infty[$.

Il est donc, théoriquement, impossible d'avoir une approximation convergente. A défaut, on peut chercher à avoir une approximation qui soit au moins conservative¹.

Pour cela, il faut tenir compte, dans le calcul de l'ensemble atteignable, de l'erreur d'approximation du champ de vecteurs f par f_h . L'idée est donc de réinjecter l'erreur d'interpolation dans l'équation différentielle sous la forme d'un terme de perturbation :

$$x'_h(t) = f_h(x_h(t)) + u(t), \quad x_h(t) \in D, \quad \|u(t)\| \leq \varepsilon(h) \quad (11.3.5)$$

où $\varepsilon(h)$ est l'erreur d'approximation donnée par les lemmes 11.1.1 ou 11.1.2.

Notons $\Phi_h(y, t)$ l'ensemble des points atteignables à l'instant t par les solutions de l'équation différentielle (11.3.5) pour la condition initiale y . L'ensemble atteignable par les solutions de l'équation différentielle (11.3.5) pour l'ensemble de valeurs initiales I est donc :

$$\mathcal{R}_h(I) = \bigcup_{y \in I, s \in \mathbb{R}^+} \Phi_h(y, s).$$

On a de manière presque immédiate :

Théorème 11.3.1

- (Approximation conservative) $\mathcal{R}(I) \subseteq \mathcal{R}_h(I)$.
- (Convergence sur tout compact) Pour tout $t \in \mathbb{R}^+$,

$$d_H\left(\bigcup_{s \in [0, t]} \Phi(I, s), \bigcup_{s \in [0, t]} \Phi_h(I, s)\right) \leq \frac{2\varepsilon(h)}{L} (e^{L|t|} - 1).$$

Preuve : Soit $x(t)$ une solution de l'équation différentielle (11.0.1),

$$\|x'(t) - f_h(x(t))\| = \|f(x(t)) - f_h(x(t))\| \leq \varepsilon(h).$$

Donc, $x(t)$ est une solution de l'inclusion différentielle (11.3.5). Ainsi, on a bien $\mathcal{R}(I) \subseteq \mathcal{R}_h(I)$.

Soient $t \in \mathbb{R}^+$, $y \in I$, $s \in [0, t]$. Soit x_h une solution de l'inclusion différentielle (11.3.5), pour la condition initiale y . Pour tout $s \in [0, t]$,

$$\|x'_h(s) - f(x_h(s))\| \leq \|x'_h(s) - f_h(x_h(s))\| + \|f_h(x_h(s)) - f(x_h(s))\| \leq 2\varepsilon(h).$$

Soit x la solution de l'équation différentielle (11.0.1), pour la condition initiale y . D'après l'inégalité fondamentale (théorème 10.2.1), on a pour tout $s \in [0, t]$,

$$\|x(s) - x_h(s)\| \leq \frac{2\varepsilon(h)}{L} (e^{L|s|} - 1) \leq \frac{2\varepsilon(h)}{L} (e^{L|t|} - 1).$$

■

¹Ce travail a été mené en collaboration avec E. Asarin et T. Dang

Ce résultat est d'une importance capitale, notamment pour ce qui concerne l'étude des attracteurs. Supposons que l'équation différentielle (11.0.1) possède un attracteur A , le théorème 11.2.2 nous dit que l'équation différentielle approchée (11.1.2) possède également un attracteur A_h et que la limite de celui-ci est incluse dans A .

Maintenant, prenons un ensemble de conditions initiales I , tel que A est inclus dans l'ensemble atteignable $\mathcal{R}(I)$. On a alors $A \subseteq \mathcal{R}_h(I)$. Voici donc un moyen de calculer une région de D contenant l'attracteur A .

En combinant ces deux approches, on arrive à réaliser une étude relativement complète de l'attracteur de l'équation différentielle.

Il reste donc à calculer l'ensemble $\mathcal{R}_h(I)$. Pour cela, remarquons que l'équation différentielle (11.3.5) définit un système hybride linéaire par morceaux soumis à des perturbations.

Ainsi, les algorithmes présentés dans les chapitres 6 et 7 peuvent être utilisés pour le calcul d'une approximation conservative de l'ensemble atteignable $\mathcal{R}_h(I)$.

Exemple 11.3.2 *Considérons l'équation de Van der Pol :*

$$\begin{cases} x_1'(t) = x_2(t) \\ x_2'(t) = x_2(t)(1 - x_1^2(t)) - x_1(t) \end{cases}$$

sur le pavé de \mathbb{R}^2 , $D = [-3, 3] \times [-3, 3]$.

Nous avons appliqué l'algorithme de calcul de l'ensemble atteignable des systèmes hybrides linéaires par morceaux à cet exemple.

Sur la figure 11.6, nous avons dessiné certaines étapes du calcul de l'ensemble atteignable ainsi que l'approximation conservative de l'ensemble atteignable $\mathcal{R}(I)$, calculée par notre algorithme. On voit que cet ensemble contient bien le cycle limite qui est un attracteur pour l'équation de Van der Pol.

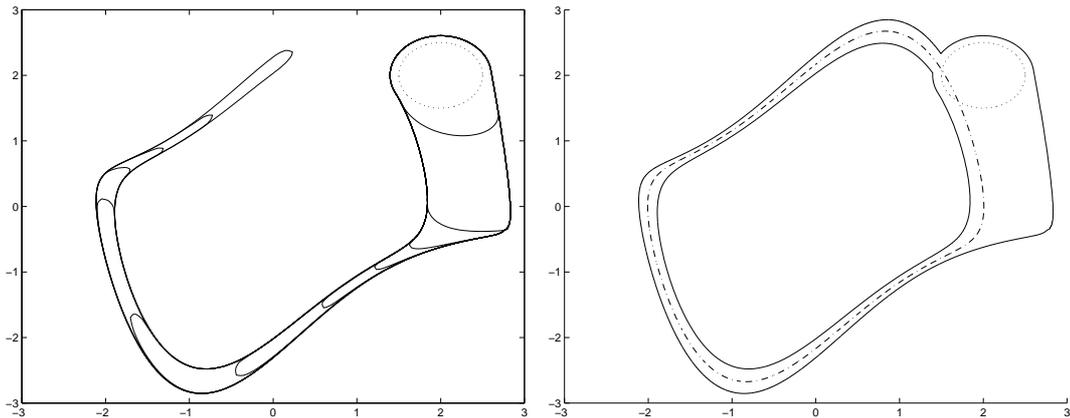


FIG. 11.6: Différentes étapes du calcul de l'ensemble atteignable de l'équation de Van der Pol pour l'ensemble de conditions initiales dessiné en pointillé (à gauche), approximation conservative de l'ensemble atteignable de l'équation de Van der Pol pour l'ensemble de conditions initiales dessiné en pointillé. Le cycle limite est contenu dans l'ensemble atteignable (à droite).

Dans ce chapitre, nous avons montré comment les méthodes de calcul hybride peuvent être utilisées pour l'étude des équations différentielles. Il apparaît que cette approche permet de développer des algorithmes efficaces qui peuvent être utilisés en complément des méthodes classiques d'analyse numérique.

Conclusion

Cette thèse est consacrée à l'analyse algorithmique des systèmes hybrides. Dans ce mémoire nous avons examiné plusieurs problèmes liés à l'étude et au contrôle des systèmes hybrides linéaires par morceaux. Les contributions dans chaque domaine sont multiples.

Dans la deuxième partie de cette thèse, nous avons considéré le problème du calcul des exécutions acceptées par un système hybride.

Dans le chapitre quatre, nous avons abordé la simulation des systèmes hybrides et plus particulièrement le problème central de la détection des événements. Nous avons proposé un algorithme pour calculer un encadrement des instants auxquels surviennent les transitions du système hybride. Cette méthode permet d'approcher ces instants avec une précision arbitraire.

Dans le chapitre cinq, nous nous sommes intéressés aux exécutions périodiques des systèmes hybrides. Notre approche utilise une extension de la notion d'application de Poincaré aux systèmes hybrides. Pour cela, nous utilisons le concept d'application d'impact, largement répandu dans la théorie des systèmes hybrides. La principale contribution de ce chapitre réside dans l'utilisation de l'application de Poincaré hybride pour le calcul des exécutions périodiques.

Dans la troisième partie, nous avons abordé le problème du calcul de l'ensemble atteignable des systèmes hybrides.

Nous avons tout d'abord exposé les deux approches existantes pour l'approximation de l'ensemble atteignable d'un système dynamique linéaire. Nous avons ensuite généralisé une de ces approches pour pouvoir considérer des systèmes dynamiques linéaires soumis à des perturbations. Nous avons proposé un algorithme permettant d'approcher l'ensemble atteignable d'un tel système par une union de polytopes et fourni une borne sur l'erreur commise par le processus d'approximation.

Enfin, nous avons intégré cette approche dans un algorithme de calcul de l'ensemble atteignable d'un système hybride.

Dans la quatrième partie, nous nous sommes intéressés au contrôle des systèmes hybrides.

Dans le chapitre huit, nous avons développé une analyse multirésolution de l'espace des entrées d'un système linéaire. Nous avons construit une base d'ondelettes orthonormales pour permettre une représentation pratique des signaux d'entrée d'un système linéaire. Cette représentation permet un traitement hiérarchique et locale de l'information.

Dans le chapitre neuf, nous avons montré comment ces propriétés pouvaient être utilisées judicieusement pour la synthèse de signaux d'entrée d'un système hybride. Nous

avons ainsi développé une méthodologie pour approcher l'entrée optimale d'un problème de contrôle d'un système hybride.

Dans la dernière partie, nous avons montré comment les techniques développées pour les systèmes hybrides linéaires par morceaux pouvaient être utilisées pour analyser des systèmes dynamiques non-linéaires.

Dans le chapitre dix, nous avons présenté la notion de calcul hybride pour montrer comment les systèmes hybrides pouvaient être utilisés pour approcher les solutions des équations différentielles scalaires.

Dans le chapitre onze, nous avons généralisé cette approche aux équations différentielles de dimensions supérieures. Nous avons proposé une méthodologie générale pour l'approximation des systèmes dynamiques par des systèmes hybrides linéaires par morceaux. Nous avons ensuite regardé plusieurs critères permettant d'évaluer la qualité de l'approximation : convergence des solutions, des attracteurs, équivalence des portraits de phase, approximation conservative de l'ensemble atteignable.

Les perspectives pour de futurs travaux sont également nombreuses.

Tout d'abord, un travail d'implémentation efficace des différentes méthodes présentées dans cette thèse est nécessaire. En effet, si tous les algorithmes présentés sont valables en dimension quelconque, ils n'ont été implémentés effectivement qu'en dimension 1 ou 2. Le passage en dimension supérieure pose des problèmes algorithmiques non-triviaux qu'il faudra gérer : représentation des polytopes et opérations en grandes dimensions, complexité exponentielle des algorithmes par rapport à la dimension du système...

Pour l'algorithme d'atteignabilité des systèmes linéaires soumis à des perturbations, on peut imaginer avoir recours à des classes particulières de polytopes pour représenter l'ensemble atteignable. Une piste est l'utilisation des zonotopes [56], qui constituent une classe de polytopes invariante pour la somme de Minkowski qui, de plus, peut être calculée avec une complexité linéaire.

Une autre piste pour les problèmes d'atteignabilité ainsi que pour l'approximation de dynamiques non-linéaires, consiste à passer d'abord par une phase de projection : il s'agit d'approcher un système dynamique par un système de dimension inférieure. Les calculs sont ensuite effectués sur le nouveau système. Des travaux récents [7, 96] nous encouragent à explorer cette voie.

Enfin, concernant la partie contrôle, nous sommes convaincus que les ondelettes de la théorie du contrôle ont de nombreuses applications dans des problèmes ne concernant pas uniquement le contrôle des systèmes hybrides. Dans [49] par exemple, nous avons déjà abordé le problème du codage efficace des entrées d'un système linéaire.

Annexe : Présentation de l'outil *CASCADE*

*CASCADE*² [51] est un outil codé en Matlab qui permet l'analyse et la simulation de systèmes dynamiques plans par des méthodes de calcul hybride présentées dans le chapitre 9.

L'utilisation de *CASCADE* est rendue très simple par l'utilisation d'une interface graphique (voir figure 11.7).

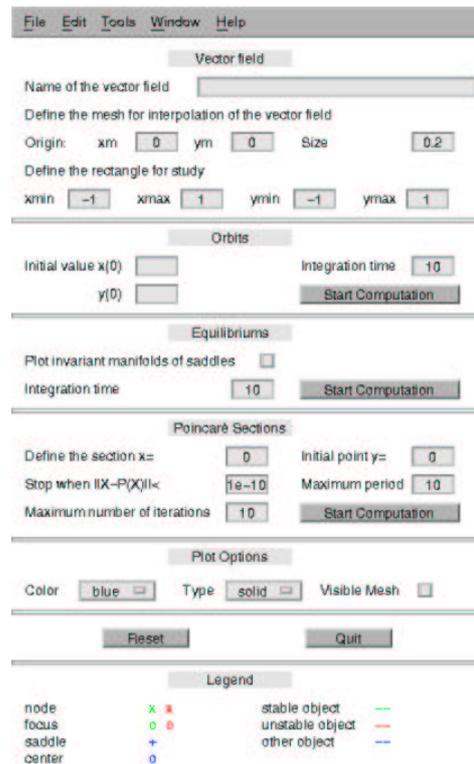


FIG. 11.7: Interface utilisateur de *CASCADE*

²Computational Analysis and Simulation using Continuous Approximations for Differential Equations

Les principales fonctionnalités de l'outil *CASCADE* sont :

– **Définition de l'approximation linéaire par morceaux du champ de vecteurs**

Le champ de vecteurs initial est rentré sous la forme d'une fonction Matlab. On définit le pavé de \mathbb{R}^2 sur lequel on souhaite étudier l'équation différentielle.

Le maillage utilisé pour l'interpolation est défini par un point qui sert d'origine et le pas de discrétisation.

– **Approximation des solutions de l'équation différentielle**

Le calcul d'une solution approchée de l'équation différentielle pour une valeur initiale donnée est effectué par l'algorithme de simulation des systèmes hybrides linéaires par morceaux présentés dans le chapitre 4.

Il faut préciser sur quel intervalle on souhaite intégrer l'équation différentielle.

– **Calcul et classification des points d'équilibre**

CASCADE classe les différents points d'équilibre en fonction des valeurs propres de la matrice du champ de vecteurs associé au domaine auquel appartient le point.

Les différentes catégories sont noeud stable, noeud instable, foyer stable, foyer instable, centre, point selle. Pour cette dernière catégorie, *CASCADE* offre la possibilité de calculer les variétés stables et instables du point d'équilibre.

– **Calcul et classification des cycles limites**

Le calcul des cycles limites est réalisé par l'algorithme de calcul des exécutions périodiques d'un système hybride linéaire par morceaux, présenté dans le chapitre 5.

Il faut préciser la droite sur laquelle on souhaite définir l'application de Poincaré, ainsi qu'un point initial sur cette droite. L'utilisateur peut préciser certains paramètres servant de critères d'arrêt de l'algorithme.

Une fois le calcul terminé, *CASCADE* détermine la stabilité du cycle limite en fonction des valeurs propres de la différentielle de l'application de Poincaré.

– **Options d'affichage**

L'utilisateur peut choisir d'afficher, ou non, le maillage servant à l'interpolation.

La majorité des figures du chapitre 9 a été dessinée grâce à l'outil *CASCADE*. L'étude d'un modèle dynamique de l'activité électrique d'un neurone, menée par J.G. Dumas et A. Rondepierre dans [33], a été partiellement réalisée par l'outil *CASCADE*.

Le code Matlab, ainsi qu'un manuel utilisateur, sont téléchargeables à l'adresse <http://www-lmc.imag.fr/MOSAIC/CASCADE/>.

Bibliographie

- [1] A.A. Andronov, A.A. Vitt, S.E. Khaikin, Theory of oscillators, Pergamon, 1966.
- [2] E. Asarin, O. Bournez, T. Dang, O. Maler, Approximate reachability analysis of piecewise linear dynamical systems, *Hybrid Systems : Computation and Control*, N. Lynch, B. H. Krogh (Eds), no. 1790 in LNCS, pp 21-31, Springer, 2000.
- [3] E. Asarin, O. Bournez, T. Dang, O. Maler, A. Pnueli, Effective synthesis of switching controllers for linear systems, *Proc. of the IEEE*, 88, Special Issue Hybrid System: Theory and Applications, 1011-1025, 2000.
- [4] E. Asarin, T. Dang, O. Maler, The d/dt tool for verification of hybrid systems. *Computer Aided Verification*, K.G. Larsen Ed Brinksma (Ed), no. 2404 in LNCS, pp 365-370, Spinger, 2002.
- [5] E. Asarin, G. Schneider, S. Yovine, Towards computing phase portraits of polygonal differential inclusions, *Hybrid Systems : Computation and Control*, C.J. Tomlin, M.R. Greenstreet (Eds), no. 2289 in LNCS, pp 49-61, Springer, 2002.
- [6] E. Asarin, T. Dang, A. Girard, Reachability of non-linear systems using conservative approximations, *Hybrid Systems : Computation and Control*, O. Maler et A. Pnueli (Eds), no. 2623 in LNCS, pp 22-35, Spinger, 2003.
- [7] E. Asarin, T. Dang, Abstraction by projection and application to multi-affine systems, *Hybrid Systems : Computation and Control*, R. Alur, G.J. Pappas (Eds), no. 2993 in LNCS, pp 32-47, Springer, 2004.
- [8] J.P. Aubin, A. Cellina, Differential inclusions, Springer-Verlag, 1984.
- [9] J.P. Aubin, Viability theory, *Systems and Control : Foundations and Applications*, Birkhäuser, 1991.
- [10] T. Bak, J. Bendtsen, A.P. Ravn, Hybrid control design for a wheeled mobile robot, *Hybrid Systems : Computation and Control*, O. Maler, Amir Pnueli (Eds), no. 2623 in LNCS, pp 50-65, Spinger, 2003.
- [11] A.M. Bayen, E. Crück, C.J. Tomlin, Guaranteed overapproximations of unsafe sets for continuous and hybrid systems: solving the Hamilton-Jacobi equation using viability techniques, *Hybrid Systems : Computation and Control*, C.J. Tomlin, M.R. Greenstreet (Eds), no. 2289 in LNCS, pp 49-61, Springer, 2002.
- [12] C. Belta, J. Schug, T. Dang, V. Kumar, M. Mintz, G.J. Pappas, H. Rubin, P. Dunlop, Stability and reachability analysis of a hybrid model of luminescence

- in the marine bacterium *Vibrio fischeri*, *Proc. IEEE Conf. on Decision and Control*, pp 869-874, 2001.
- [13] A. Bemporad, M. Morari, Control of systems integrating logic, dynamics, and constraints, *Automatica*, 35(3):407-427, 1999.
- [14] A. Bemporad, F. Borrelli, M. Morari, On the optimal control law for linear discrete time hybrid systems, *Hybrid Systems : Computation and Control*, C.J. Tomlin, M.R. Greenstreet (Eds), no. 2289 in LNCS, pp 164-178, Springer, 2002.
- [15] A. Bemporad, A. Giua, C. Seatzu, A master-slave algorithm for the optimal control of continuous-time switched affine systems. *Proc. IEEE Conf. on Decision and Control*, 2002.
- [16] A. Bemporad, P. Borodani, M. Manneli, Hybrid control of an automotive robotized gearbox for reduction of consumptions and emissions, *Hybrid Systems : Computation and Control*, O. Maler, Amir Pnueli (Eds), no. 2623 in LNCS, pp 81-96, Springer, 2003.
- [17] M.S. Branicky, V.S. Borkar, S.K. Mitter, A unified framework of hybrid control : model and optimal control theory, *IEEE Trans. Automatic Control*, 43(1):31-45, 1998.
- [18] B. Bueler, A. Enge, K. Fukuda, Exact volume computation for polytopes: a practical study, *Polytopes-combinatorics and computation*, G. Kalai, G.M. Ziegler (Eds), DMV-Seminars vol. 29, Birkhäuser Verlag, Basel 2000.
- [19] L.O. Chua, A.C. Dang, Canonical piecewise-linear modeling, *IEEE Trans. Circuits and Systems*, 33(5):511-525, 1986.
- [20] L.O. Chua, T.S. Parker, Practical numerical algorithms for chaotic systems, Springer , 1989.
- [21] A. Chutinan, B.H. Krogh, Verification of polyhedral invariant hybrid automata using polygonal flow pipe approximations. *Hybrid Systems : Computation and Control*, F. Vaandrager, J. van Schuppen (Eds), no. 1569 in LNCS, pp 76-90, Springer, 1999.
- [22] A. Chutinan, A. Fehnker, Z. Han, J. Kapinski, R. Kumar, B. H. Krogh, O. Stursberg, CheckMate, hybrid systems verification toolbox for MATLAB, <http://www.ece.cmu.edu/~webk/checkmate/>
- [23] T. Dang, O. Maler, Reachability analysis via face lifting, *Hybrid Systems : Computation and Control*, T.A. Henzinger, S. Sastry (Eds), no. 1386 in LNCS, pp 96-109, Springer, 1998.
- [24] T. Dang, Vérification et synthèse des systèmes hybrides, *Thèse de Doctorat*, Institut National Polytechnique de Grenoble, Grenoble, 2000.
- [25] T. Dang, O. Maler, d/dt, reachability analysis of continuous and hybrid systems, <http://www-verimag.imag.fr/~tdang/ddt.html>.
- [26] I. Daubechies, Ten lectures on wavelets, *CBMS-NSF Regional Conference Series in Applied Mathematics*, SIAM, 1992.
- [27] D.L. Delchamps, State space and input-output linear systems, Springer, 1988.
- [28] J. Della Dora, A. Maignan, M. Mirica-Ruse, S. Yovine, Hybrid computation, *Proc. ISSAC'01*, pp 101-108, 2001.

-
- [29] J.P. Demailly, Analyse numérique et équations différentielles, *Collection Grenoble Sciences*, Presses Universitaires de Grenoble, 1996.
- [30] J. Dieudonné, Calcul infinitésimal, *Collection Méthodes*, Hermann, 1980.
- [31] A.L. Dontchev, I.V. Kolmanovskiy, State constraints in the linear regulator problem : case study, *J. Optimization Theory and Applications*, 87(2):323-347, 1995.
- [32] J.R. Dormand, Numerical methods for differential equations, *Library of Engineering Mathematics*, CRC Press, 1996.
- [33] J.G. Dumas, A. Rondepierre, Modeling the electrical activity of a neuron by a continuous and piecewise linear hybrid system, *Hybrid Systems : Computation and Control*, O. Maler, Amir Pnueli (Eds), no. 2623 in LNCS, pp 22-35, Springer, 2003.
- [34] M. Egerstedt, P. Ögren, O. Shakernia, J. Lygeros, Toward optimal control of switched linear systems, *Proc. IEEE Conf. on Decision and Control*, 2000.
- [35] M. Egerstedt, C.F. Martin, Optimal trajectory planning and smoothing splines, *Automatica*, 37:1057-1064, 2001.
- [36] M. Egerstedt, C.F. Martin, Optimal control and monotone smoothing splines, *New Trends in Nonlinear Dynamics and Control, and their Applications*, no. 295 in LNCIS, pp 279-294, Springer, 2003.
- [37] N.H. El-Farra, P.D. Christofides, Switching and feedback laws for control of constrained switched nonlinear systems, *Hybrid Systems : Computation and Control*, C.J. Tomlin, M.R. Greenstreet (Eds), no. 2289 in LNCS, pp 164-178, Springer, 2002.
- [38] J.M. Esposito, V. Kumar, G.J. Pappas, Accurate event detection for simulating hybrid systems, *Hybrid Systems : Computation and Control*, M.D.D. Benedetto, A.L. Sangiovanni-Vincentelli (Eds), no. 2034 in LNCS, pp 204-217, Springer, 2001.
- [39] G. Ewald, Combinatorial convexity and algebraic geometry, *Graduate Texts in Mathematics*, Springer, 1996.
- [40] G. Ferrari-Trecate, F.A. Cuzzola, M. Morari, Analysis of discrete-time piecewise affine systems with logic states, *Hybrid Systems : Computation and Control*, C.J. Tomlin, M.R. Greenstreet (Eds), no. 2289 in LNCS, pp 164-178, Springer, 2002.
- [41] A.F. Filippov, Differential equations with discontinuous right-hand side, Kluwer Academic Publishers, 1988.
- [42] H. Freudenthal, Simplicialzerlegungen von beschraeukter flachkeit. *Annals of Math. in Science and Engin.*, 43:580-582, 1942.
- [43] M. Garavello, B. Piccoli, Hybrid necessary principles: an application to a car with gears, *Proc. IFAC Conf. ADHS 2003*, Elsevier, 2003.
- [44] R. Ghosh, A. Tiwari, C. Tomlin, Automated symbolic reachability analysis; with application to Delta-Notch signaling automata. *Hybrid Systems : Computation and Control*, O. Maler, Amir Pnueli (Eds), no. 2623 in LNCS, pp 22-35, Spinger, 2003.

-
- [45] A. Girard, Analysis of ODE using hybrid computation, *Rapport Technique IMAG*, RT 181, 2002.
- [46] A. Girard, Approximate solutions of ODEs using piecewise linear vector fields, *Proc. CASC 2002*, pp 107-120, 2002.
- [47] A. Girard, Detection of event occurrence in piecewise linear hybrid systems, *Proc. RASC 2002*, pp 19-25, 2002.
- [48] A. Girard, Computation and stability analysis of limit cycles in piecewise linear hybrid systems, *Proc. IFAC Conf. ADHS 2003*, Elsevier, 2003.
- [49] A. Girard, Towards a multiresolution approach to linear control, *IEEE Trans. Automatic Control*, accepté, 2004.
- [50] A. Girard, Optimal control of linear systems : a multiresolution approach, *Proc. IEEE Conf. on Decision and Control*, à paraître, 2004.
- [51] A. Girard, *CASCADE* : Computational Analysis and Simulation using Continuous Approximations for Differential Equations. <http://www-lmc.imag.fr/MOSAIC/CASCADE/>.
- [52] J.M. Goncalves, A. Megretski, M.A. Dahleh, Global analysis of piecewise linear systems using impact maps and surface Lyapunov functions, *IEEE Trans. Automatic Control*, 48(12):2089-2106, 2003.
- [53] J.E. Goodman, J. O'Rourke, Handbook of discrete and computational geometry, CRC Press , 1997.
- [54] M.R. Greenstreet, I. Mitchell, Reachability analysis using polygonal projections, *Hybrid Systems : Computation and Control*, F. Vaandrager, J. van Schuppen (Eds), no. 1569 in LNCS, pp 76-90, Springer, 1999.
- [55] J. Guckenheimer, B. Meloon, Computing periodic orbits and their bifurcations with automatic differentiation. *SIAM Journal of Scientific Computing*, 22:951-985, 2000.
- [56] L.J. Guibas, A. Nguyen, L. Zhang, Zonotopes as Bounding Volumes, *Proc. ACM-SIAM symposium on discrete algorithms*, 2003.
- [57] W.P.M.H. Heemels, M.K. Camlibel, A.J. van der Schaft, J.M. Schumacher, Modelling, well-posedness, and stability of switched electrical networks, *Hybrid Systems : Computation and Control*, O. Maler, Amir Pnueli (Eds), no. 2623 in LNCS, pp 22-35, Springer, 2003.
- [58] T.A. Henzinger, P.H. Ho, H. Wong-Toi, HyTech : A model checker for hybrid systems. *Software Tools for Technology Transfer*, 1:110-122, 1997.
- [59] I.A. Hiskens, M.A. Pai, Trajectory sensitivity analysis of hybrid systems, *IEEE Trans. on Circuits and Systems*, 47(2):204-220, 2000.
- [60] I.A. Hiskens, Stability of hybrid system limit cycles : application to the compass gait biped robot. *Proc. IEEE Conf. on Decision and Control*, 774-779, 2001.
- [61] I.A. Hiskens, Stability of limit cycles in hybrid systems, *Proc. Hawaii Int. Conf. on System Sciences*, 2001.
- [62] J. Hubbard, B. West, Differential equations : a dynamical system approach, higher dimensional systems, *Texts in Applied Mathematics*, Vol. 18, Springer-Verlag, 1995.

-
- [63] M. Johansson, A. Rantzer, Computation of piecewise quadratic Lyapunov functions for hybrid systems, *IEEE Trans. Automatic Control*, 43(4):555-559, 1998.
- [64] K.H. Johansson, J. Lygeros, S. Sastry, M. Egerstedt, Simulation of Zeno hybrid automata, *Proc. IEEE Conf. on Decision and Control*, 1999.
- [65] K.H. Johansson, M. Egerstedt, J. Lygeros, S. Sastry, On the regularization of Zeno hybrid automata, *Systems and Control Letters*, 38:141-150, 1999.
- [66] H.W. Kuhn, Some combinatorial lemmas in topology, *IBM Journal of Research and Development*, 4(5):518:524, 1960.
- [67] G. Lafferriere, G. Pappas, S. Yovine, Reachability computation for linear systems, *Proc. IFAC World Congress*, E:7-12, 1999.
- [68] B. Lincoln, A. Rantzer, Relaxed optimal control of piecewise linear systems *Proc. IFAC Conf. ADHS 2003*, Elsevier, 2003.
- [69] J. Llibre, E. Nunez, A.E. Teruel, Limit cycles for planar piecewise linear differential systems via first integrals. *Qualitative Theory of Dynamical Systems*, 3(1):29-50, 2002.
- [70] J. Lygeros, K.H. Johansson, S. Sastry, M. Egerstedt, On the existence of executions of hybrid automata, *Proc. IEEE Conf. on Decision and Control*, 1999.
- [71] J. Macki, A. Strauss, Introduction to optimal control theory, Springer, 1982.
- [72] S. Mallat, A wavelet tour of signal processing, *Academic Press*, 1998.
- [73] A.S. Matveev, A.V. Savkin, Qualitative theory of hybrid dynamical systems, Birkhäuser, 2000.
- [74] M. Mirica-Ruse, Contribution à l'étude des systèmes hybrides, *Thèse de Doctorat*, Université Joseph Fourier, Grenoble, 2002.
- [75] I. Mitchell, C. Tomlin, Level set method for computation in hybrid systems, *Hybrid Systems : Computation and Control*, N. Lynch, B.H. Krogh (Eds), no. 1790 in LNCS, pp 421-436, Springer, 2000.
- [76] T. Park, P.I. Barton, State event location in differential-algebraic models, *ACM Trans. Modeling and Computer Simulation*, 6(2):137-165, 1996.
- [77] N.B.O.L. Pettit, P.E. Wellstead, Analysing piecewise linear dynamical systems, *IEEE Control Systems Magazine*, 15(5):43-50, 1995.
- [78] L. Perko, Differential equations and dynamical systems, *Texts in Applied Mathematics*, Vol. 7, Springer-Verlag, 1991.
- [79] L.S. Pontryagin, Ordinary differential equations, A.W., Reading, Mass, 1962.
- [80] M. Rubensson, B. Lennartson, S. Pettersson, Convergence to limit cycles in hybrid systems : an example, *Large Scale Systems : Theory and Applications*, 704-709, 1998.
- [81] M. Rubensson, B. Lennartson, Stability of limit cycles in hybrid systems using discrete-time Lyapunov techniques. *Proc. IEEE Conf. on Decision and Control*, 1397-1402, 2000.
- [82] A.M. Samoilenko, N.A. Perestyuk, Impulsive differential equations, *Series on Nonlinear Science*, World Scientific, 1995.

-
- [83] M. Senesky, G. Eirea, T.J. Koo, Hybrid modelling and control of power electronics, *Hybrid Systems : Computation and Control*, O. Maler, A. Pnueli (Eds), no. 2623 in LNCS, pp 22-35, Springer, 2003.
- [84] M.S. Shaikh, P.E. Caines, On the optimal control of hybrid systems : optimization of trajectories, switching times and location schedule, *Hybrid Systems : Computation and Control*, O. Maler, A. Pnueli (Eds), no. 2623 in LNCS, pp 22-35, Springer, 2003.
- [85] L.F. Shampine, I. Gladwell, R.W. Brankin, Reliable solution of special event location problems for ODEs, *ACM Trans. Mathematical Software*, 17(1):11-25, 1991.
- [86] B. I. Silva, K. Richeson, B. H. Krogh, A. Chutinan, Modeling and verification of hybrid dynamical system using CheckMate, *Proc. Int. Conf. on Automation of Mixed Processes ADPM*, 2000.
- [87] S. Simic, K.H. Johansson, S. Sastry, J. Lygeros, Towards a geometric theory of hybrid systems, *Hybrid Systems : Computation and Control*, N. Lynch, B. H. Krogh (Eds), no. 1790 in LNCS, pp 421-436, Springer, 2000.
- [88] S. Simic, K.H. Johansson, J. Lygeros, S. Sastry, Hybrid limit cycles and hybrid Poincaré-Bendixon, *IFAC world congress*, 2002.
- [89] A.M. Stuart, A.R. Humphries, Dynamical systems and numerical analysis, *Cambridge monographs on applied and computational mathematics*, Cambridge University Press , 1996.
- [90] H. Sussman, A maximum principle for hybrid control problems, *Proc. IEEE Conf. on Decision and Control*, pp 425-430, 1999.
- [91] S. Sun, B. Egerstedt, C.F. Martin, Control theoretic smoothing splines, *IEEE Trans. Automatic Control*, 45(2):2271-2279, 2000.
- [92] W. Sweldens, The lifting scheme : a construction of second generation wavelets, *SIAM J. Mathematical Analysis*, 29(2):511-546, 1997.
- [93] C. Tomlin, G.J. Pappas, J. Lygeros, D.N. Godbole, S. Sastry, Hybrid control models of next generation air traffic management, *Hybrid Systems IV*, no 1273 in LNCS, Springer 1997.
- [94] R.J. Vanderbei, LOQO : an interior point code for quadratic programming, *Optimization Methods and Software*, 11:451-484, 1999.
- [95] A. van der Schaft, H. Schumacher, An introduction to hybrid dynamical systems, no. 251 in LNCIS, Springer, 2000.
- [96] A. van der Schaft, Bisimulation of dynamical systems, *Hybrid Systems : Computation and Control*, R. Alur, G.J. Pappas (Eds), no. 2993 in LNCS, pp 555-569, Springer, 2004.
- [97] D. Viswanath, The Lindstedt-Poincaré technique as an algorithm for computing periodic orbits, *SIAM Review*, 43(3):478-495, 2001.
- [98] S. Wiggins, Introduction to applied nonlinear dynamical systems and chaos, *Texts in Applied Mathematics*, Vol. 2, Springer-Verlag, 1995.
- [99] S. Yovine, Kronos : a verification tool for real-time systems, *Software Tools for Technology Transfer*, 1:123-133, 1997.

- [100] J. Zhang, K.H. Johansson, J. Lygeros, S. Sastry, Zeno hybrid systems, *International Journal of Robust and Nonlinear Control*, 11(5):435-451, 2001.
- [101] G.M. Ziegler, Lectures on polytopes, *Graduate Texts in Mathematics*, Springer-Verlag, 1995.

Résumé : Cette thèse est consacrée à l'analyse algorithmique des systèmes hybrides. Nous examinons plusieurs problèmes liés à l'étude et au contrôle des systèmes hybrides linéaires par morceaux. Dans une première partie, nous présentons les notions de base de la théorie. Nous illustrons notre propos grâce à de nombreux exemples. La deuxième partie est dédiée au calcul algorithmique des exécutions acceptées par un système hybride. Une méthode de détection des événements (changement de valeur de la variable discrète du système) est proposée. Le cas des exécutions périodiques est également examiné. Dans la troisième partie, nous abordons le problème du calcul de l'ensemble atteignable des systèmes hybrides. Nous apportons un soin particulier aux systèmes où les dynamiques continues sont connues de manière incertaine. Dans la quatrième partie, nous nous intéressons au contrôle des systèmes hybrides. Nous construisons une analyse multirésolution de l'espace des entrées d'un système linéaire et calculons une base d'ondelettes associée. Les propriétés de cette base se révèlent intéressantes pour la synthèse de signaux d'entrée d'un système hybride. Dans la dernière partie nous montrons que les techniques développées pour les systèmes hybrides linéaires par morceaux peuvent être utilisées pour analyser des systèmes dynamiques non-linéaires.

Mots-clés : systèmes hybrides, équations différentielles linéaires, simulation, exécution périodique, atteignabilité, approximation, contrôle optimal, ondelettes.

Abstract : This thesis is devoted to the algorithmic analysis of hybrid systems. We consider several problems linked to the study and the control of piecewise linear hybrid systems. In the first part, we introduce the fundamental notions of the theory. These are illustrated by numerous examples. The second part is devoted to the computation of the executions accepted by a hybrid system. A method for event detection is proposed. The case of periodic executions is also examined. In the third part, we consider the problem which consists in computing the reachable set of a hybrid system. Particularly, we propose a method for the systems where the continuous dynamics are given by uncertain differential equations. In the fourth part, we propose an approach for the control of hybrid systems. We construct a multiresolution analysis of the set of inputs of a linear system and compute an associated set of wavelets. The properties of this basis are interesting for the synthesis of input signals of a hybrid system. In the last part, we show that the methods developed for piecewise linear hybrid systems can be used to analyse non-linear dynamical systems.

Keywords : hybrid systems, linear differential equation, simulation, periodic execution, reachability, approximation, optimal control, wavelets.