Computational Approaches to Analysis and Control of Hybrid Systems

Antoine Girard

UNIVERSITÉ DE GRENOBLE

HABILITATION À DIRIGER DES RECHERCHES

Spécialité : **Mathématiques Appliquées**

Ecole Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique

Présentée par

Antoine Girard

préparée au sein du **Laboratoire Jean Kuntzmann**

Computational Approaches to Analysis and Control of Hybrid Systems

Habilitation à diriger des recherches soutenue publiquement
le **19 novembre 2013**,
devant le jury composé de :

M. Bernard Brogliato
Directeur de Recherches, INRIA, Président
M. Jean-Michel Coron
Professeur, Université Paris 6, Rapporteur
M. Bruce Krogh
Professeur, Carnegie Mellon University, USA, Rapporteur
M. John Lygeros
Professeur, ETH Zurich, Suisse, Rapporteur
M. Jamal Daafouz
Professeur, Université de Lorraine, Examinateur
M. Oded Maler
Directeur de Recherches, CNRS, Examinateur
M. Emmanuel Trélat
Professeur, Université Paris 6, Examinateur

Remerciements

Je tiens d'abord à remercier chaleureusement les membres du jury et particulièrement Jean-Michel Coron, Bruce Krogh et John Lygeros pour leur travail de rapporteurs. Merci à Jamal Daafouz, Oded Maler et Emmanuel Trélat pour leur rôle d'examinateurs et à Bernard Brogliato pour avoir accepté de présider ce jury.

Ce mémoire présente mes travaux de recherches depuis ma thèse préparée sous la direction de Jean Della Dora et soutenue en septembre 2004. Je souhaite ici lui rendre hommage, son influence sur la manière d'appréhender le travail de chercheur aura été déterminante; j'espère qu'il aurait été fier du travail accompli.

Je souhaite également remercier mes collègues du Laboratoire Jean Kuntzmann ainsi que tous mes collaborateurs. Je remercie les étudiants (doctorants ou post-doctorants) qui ont choisi de travailler avec moi: Colas Le Guernic, Samuel Martin, Amin Ben Sassi, Pierre-Olivier Lamare, Pierre-Jean Meyer, Ying Tang, Gang Zheng, Constantin Morarescu, Javier Camara, Sebti Mouelhi, Euriell Le Corronc. Je remercie aussi Oded Maler, Guillaume James, Christophe Prieur, Emmanuel Witrant et Gregor Goessler pour leur travail de co-encradrement.

Enfin, mes remerciements vont à ma famille et en particulier à mon épouse Anne pour son soutien indéfectible au cours de toutes ces années. Merci à Arthur d'avoir bien voulu attendre que je finisse de rédiger ce mémoire pour arriver et pour tout le bonheur qu'il nous a apporté depuis.

Résumé

Contexte Scientifique

Un *système hybride* est un système dynamique exhibant à la fois des comportements de nature discrète et continue. Motivée par la multiplication de composants informatiques embarqués "discrets" interagissant avec le monde physique "continu" (un domaine d'application aujourd'hui dénommé systèmes cyber-physiques), la recherche sur les systèmes hybrides s'est développée rapidement depuis les années 90 à l'intersection de l'informatique, de l'automatique et des mathématiques appliquées. Chacune de ces disciplines a apporté ses propres modèles et méthodes et leur confrontation et combinaison ont permis à la communauté d'établir les fondations d'une théorie des systèmes hybrides. La notion d'automate hybride [Hen96, LJS+03], qui constitue le modèle mathématique de systèmes hybrides le plus couramment utilisé, combine équations différentielles et automates d'états finis, et constitue un exemple typique de cette fertilisation croisée. Plus généralement, la recherche sur les systèmes hybrides a permis le développement d'approches nouvelles en informatique et en contrôle qui n'aurait pas été possible sans les interactions fortes entre disciplines. Le domaine du *contrôle symbolique* [EFE06, Tab09] par exemple a emprunté à l'informatique des outils et des concepts tels que la vérification formelle, l'analyse d'atteignabilité, l'abstraction ou la logique, et les a appliqués à la conception de systèmes de commande. Dans le contrôle symbolique, les dynamiques continues sont abstraites sur un ensemble fini de symboles, chaque symbole représentant une infinité d'états. Ces approches permettent de prendre en compte des spécifications qui sont souvent différentes des propriétés traditionnelles en théorie du contrôle (e.g. stabilité, contrôlabilité, observabilité...): de telles spécifications peuvent par exemple être données par des formules logiques décrivant les comportements temporels acceptables du système. Néanmoins, les approches fructueuses empruntent souvent autant à l'informatique qu'à l'automatique ou aux mathématiques appliquées (utilisation de fonctions de Lyapunov, approximations numériques...). Finalement, le contrôle symbolique accorde aussi une place importante aux algorithmes et au développement de techniques computationnelles pour l'analyse et le contrôle des systèmes dynamiques.

La plus grande partie de mon travail de recherche appartient au domaines des systèmes hybrides et du contrôle symbolique, avec une attention particulière portée au développement de techniques computationnelles. Une partie de mon travail porte également sur l'analyse des systèmes dynamiques multi-agents. Ce mémoire présente les contributions principales de mon travail de recherche depuis ma thèse en 2004. La présentation n'est pas exhaustive mais s'attache à décrire les résultats que je considère comme étant les plus significatifs. Je mets également en avant un certain nombre de résultats qui ont été obtenus en collaboration avec des jeunes chercheurs, doctorants ou post-doctorants que j'ai supervisés. Une

liste complète de mes publications peut être trouvée en annexe.

Contributions Principales

Ce mémoire est organisé en trois parties principales. La première partie introduit un cadre d'approximation qui s'applique aux systèmes dynamiques continus, discrets et hybrides (Chapitre 2); plusieurs applications de ce cadre sont présentées par la suite (Chapitres 3 et 4). La deuxième partie est consacrée à l'analyse d'atteignabilité (Chapitre 5), une technique computationnelle très utile pour l'analyse et le contrôle des systèmes hybrides. Enfin, la troisième partie porte sur les systèmes dynamiques multi-agents (Chapitre 6). Nous décrivons ici brièvement chaque partie.

Simulation et bisimulation approchées

Les théories d'approximation sont fondamentales pour l'analyse et le contrôle de systèmes dynamiques complexes. Pour des systèmes dynamiques continus, la notion d'approximation est souvent caractérisée au travers de métriques mesurant la distance entre les comportement de deux systèmes (voir e.g. [ASG00]). Pour des systèmes discrets, où une notion naturelle de distance entre les comportements n'est pas toujours disponible, la notion d'approximation est généralement appréhendée par des relations d'ordre ou d'équivalence telle que l'inclusion de langage, les relations de simulation ou de bisimulation [Mil89, CGP00].

Un défi majeur de la théorie des systèmes hybrides est de proposer un cadre commun pour l'approximation de dynamiques continus, discrètes et hybrides. Plusieurs travaux ont par exemple étendu la notion de relations de simulation et bisimulation aux systèmes continus et hybrides [Pap03, vdS04, HTP05]. Dans [GP07b], nous avons introduit les notions d'inclusion approchée de langage, de relations de simulation et bisimulation approchées et nous avons défini une hiérarchie associée de métriques d'approximation pour des systèmes (continus, discrets ou hybrides) observés sur des espaces métriques[1]. Intuitivement, ces métriques mesurent la qualité d'approximation d'un système par un autre en se basant sur la distance entre leurs comportements observés; les notions d'inclusion "exacte" de langage, de simulation et de bisimulation étant retrouvées lorsque les métriques s'annulent. Une caractérisation fonctionnelle de la simulation et bisimulation approchées a été établie en introduisant la notion de fonctions de simulation et bisimulation qui sont définies par de inégalités variationnelles de type Lyapunov.

La bisimulation approchée s'est avérée être un outil puissant pour la synthèse de contrôleur basée sur l'abstraction. Dans [Gir12], nous avons présenté des approches pour la synthèse de contrôleurs de sûreté et d'atteignabilité utilisant des abstractions approximativement bisimilaires. Etant donné un contrôleur pour l'abstraction, nous pouvons en déduire un contrôleur pour le système original en utilisant des procédures de concrétisation spécifiques. La relation de bisimulation approchée entre le système et son abstraction nous permet de garantir que le contrôleur est "correct par construction", ce qui signifie que la spécification est vérifiée par le système original. De plus, les performances du contrôleur maximal (pour la sûreté) et optimal (pour l'atteignabilité) peuvent être approchées arbitrairement près en utilisant des abstractions suffisamment

[1] Cet article s'est vu attribué le George S. Axelby Outstanding Paper Award décerné par l'IEEE Control System Society en 2009.

précises. Ces approches peuvent être utilisées pour la synthèse de contrôleurs pour des classes de systèmes dynamiques (continus ou hybrides) incrémentalement stables, pour lesquelles nous avons démontré l'existence d'abstractions approximativement bisimilaires de précision arbitraire [Gir07, PGT08, GPT10]. Comme ces abstractions sont symboliques, cela nous permet de recourir aux techniques développées dans le domaine des systèmes discrets pour contrôler des systèmes continus ou hybrides. Nous avons proposé des techniques pour déduire la complexité des contrôleurs symboliques résultants, en utilisant une quantification de l'état et une représentation efficace de la fonction de contrôle [Gir13]. Nous avons aussi développé une approche pour réduire la complexité algorithmique de la synthèse de contrôleur en utilisant des abstractions symboliques multi-échelles [CGG11b, CGG11a]. Les algorithmes de synthèse exploitant les spécificités de ces abstractions ont été implémentés dans l'outil CoSyMA [MGG13]: les abstractions sont calculées à la volée durant la synthèse de contrôleurs et la dynamique aux échelles les plus fines n'est explorée que lorsque cela est nécessaire. Les résultats expérimentaux montrent une réduction significative du coût algorithmique de la synthèse de contrôleurs.

Nous avons exploré des applications de la simulation et bisimulation approchées en dehors du domaine du contrôle symbolique mentionné précédemment. Par exemple, la notion de fonction de simulation a été utilisée pour relier formellement les comportements de deux systèmes continus afin de concevoir des contrôleurs hiérarchiques [GP09]. Nous avons établi des caractérisations des relations de simulation et de bisimulation approchées afin de calculer des approximations de systèmes continus ou hybrides [GP07a, GJP08]. Enfin, les fonctions d'auto-bisimulation (fonctions de bisimulation entre un système et lui même) ont été utilisées dans des algorithmes de vérification qui peuvent déterminer qu'une propriété est vérifiée par une infinité de trajectoires en ne simulant qu'un nombre fini d'entre-elles [FGP06].

Les résultats mentionnés ci-dessus ont été développés durant mon séjour post-doctoral à l'Université de Pennsylvanie puis dans le cadre des projets ANR VAL-AMS et VEDECY et du projet SYMBAD du pôle MSTIC de l'Université Joseph Fourier. Une grande partie de ces résultats a été développée en collaboration avec George J. Pappas (Université de Pennsylvanie), Paulo Tabuada (UCLA), Giordano Pola (Université de L'Aquila) ou Gregor Goessler (INRIA) pour ne citer que les collaborations les plus suivies. Enfin, le travail sur les abstractions symboliques multi-échelles a été réalisé durant les séjours post-doctoraux de Javier Càmara et Sebti Mouelhi à l'INRIA sous la co-supervision de Gregor Goessler et moi-même.

Analyse d'atteignabilité

L'analyse d'atteignabilité est une problématique majeure de la recherche sur les systèmes hybrides. Cette approche, inspirée par des idées de la vérification des systèmes discrets et de la simulation numérique des systèmes continus, cherche à calculer (une approximation de) l'ensemble des trajectoires d'un système, pour toutes valeurs admissibles des conditions initiales et des paramètres, et sous toutes les perturbations. L'analyse d'atteignabilité permet donc de remplacer une infinité de simulations de trajectoires individuelles [Mal11]. De plus, les algorithmes pour l'analyse d'atteignabilité sont au coeur de plusieurs approches computationnelles pour résoudre des problèmes du domaine des systèmes hybrides tels que la synthèse de contrôleur, la vérification ou le calcul d'abstractions symboliques [ABD+00, TMBO03, ADI06]. Pour que ces approches soient mathématiquement

rigoureuses, il est souvent nécessaire de garantir des propriétés de l'approximation calculée. Par exemple, il est souvent requis que l'approximation calculée contienne le véritable ensemble atteignable. Ainsi, mes contributions à ce problème concernent principalement les algorithmes pour calculer des sur-approximations de l'ensemble des états atteignables par un système dynamique continu.

Pour des systèmes linéaires avec des valeurs d'entrées bornées, nous avons proposé des schémas de discrétisation en temps nous permettant de calculer une sur-approximation de l'ensemble des états atteignables sur un intervalle de temps borné [Gir05, LG10]. Cette approximation consiste en une union finie d'ensembles convexes compacts et peut être choisie de précision arbitraire en réduisant le pas de temps. Nous avons développé plusieurs implémentations de ces schémas d'approximation. Le premier est basé sur les zonotopes [Gir05], une classe de polytopes avec des propriétés computationelles intéressantes. Une amélioration déterminante de cette approche, basée sur l'introduction de séquences d'ensembles auxiliaires, a été présentée dans [GLM06], ouvrant la voie au calcul d'approximations très précises y compris en grande dimension. Cette technique a été étendue pour travailler avec des ensembles convexes compacts arbitraires en proposant une implémentation basée sur les fonctions support [LG10]. L'adaptation de ces algorithmes pour l'analyse d'atteignabilité des automates hybrides a été décrite dans [GL08, LG09]. De manière remarquable, l'algorithme basé sur les fonctions support est aujourd'hui au coeur de la plateforme de vérification des systèmes hybrides, SpaceEx [FLD+11], développée au laboratoire Verimag.

Nous avons aussi travaillé sur l'analyse d'atteignabilité des systèmes non-linéaires. Pour les systèmes dynamiques polynomiaux en temps discret, nous avons développé une approche pour le calcul de sur-approximations polytopiques de l'ensemble atteignable en utilisant des relaxations linéaires de problèmes d'optimisation polynomiale. Ces programmes linéaires sont obtenus grâce aux propriétés des polynômes de Bernstein [BTDG12]. Pour les systèmes non-linéaires généraux, l'analyse d'atteignabilité peut être attaquée grâce au principe d'hybridation [ADG03, ADG07], qui consiste à approcher la dynamique non-linéaire par une dynamique linéaire par morceaux où des perturbations bornées sont ajoutées pour prendre en compte l'erreur d'approximation. L'ensemble atteignable de l'automate hybride résultant, calculable par les approches mentionnées ci-dessus, fournit un sur-approximation de l'ensemble atteignable du système non-linéaire original.

Une partie des travaux décrits sur l'analyse d'atteignabilité a été réalisée dans le cadre du projet ANR VEDECY. Ces travaux sont le résultat d'une collaboration suivie avec les chercheurs de Verimag, Thao Dang, Oded Maler et Goran Frehse. L'analyse d'atteignabilité des systèmes linéaires a été amplement traitée dans le cadre de la thèse de Colas Le Guernic sous la co-supervision d'Oded Maler et moi-même. L'analyse des systèmes polynomiaux constitue une des applications considérée dans la thèse de Mohamed Amin Ben Sassi sous la co-supervision de Guilaume James et moi-même.

Systèmes dynamiques multi-agents

Une problématique centrale dans les systèmes dynamiques multi-agents consiste en la compréhension de l'émergence de comportements collectifs coordonnés à partir de simples règles d'interaction entre agents. Le problème le plus étudié dans ce domaine est le problème du consensus où les agents cherchent à atteindre un accord de manière distribuée. Les algorithmes de consensus ont plusieurs applications dans des domaines aussi

différents que le contrôle de flottes de véhicules en robotique [RBA07] ou la modélisation de dynamiques d'opinions en sciences sociales [HK02].

Des conditions suffisantes pour la convergence vers le consensus sont typiquement basées sur la topologie du réseau décrivant les interactions entre agents et sur la force de ces interactions. Dans [MG13], nous avons établi un ensemble de conditions suffisantes pour atteindre le consensus en temps continu. Celles-ci requièrent principalement que la connectivité du graphe d'interaction soit persistante et que la divergence entre les poids d'interactions réciproques ne soit pas trop rapide. Nous avons aussi établi une estimation du taux de convergence vers le consensus et montré que notre résultat généralise plusieurs conditions existantes dans la littérature, y compris celles établies récemment dans [HT12].

Nous avons aussi proposé et étudié un modèle de dynamique d'opinions qui peut expliquer la formation de communautés au sein d'un réseau d'agents [MG11]. Dans notre modèle, les agents cherchent à atteindre le consensus avec un taux de convergence contraint. Cela peut être vu comme un processus de négociation où chaque agent exige, pour poursuivre la négociation, que les autres approches suffisamment rapidement son opinion. Dans ce cas, le consensus n'est peut être pas réalisé globalement mais uniquement dans certains sous-groupes d'agents que nous appelons communautés. Nous avons établi une caractérisation de ces communautés en terme de propriétés algébriques du graphe d'interactions. De plus, nous avons pu vérifier expérimentalement, sur un certain nombre de cas d'étude, que ces communautés coïncident avec celles observées dans la réalité. Ainsi, notre modèle fournit une solution naturellement distribuée au problème de détection de communautés dans des réseaux complexes.

Les conditions suffisantes pour la convergence vers le consensus constituent la principale contribution de la thèse de Samuel Martin sous la co-supervision de Guillaume James et moi-même. Le modèle de dynamique d'opinions a été étudié en collaboration avec Constantin Morarescu durant son séjour post-doctoral au Laboratoire Jean Kuntzmann sous ma supervision, dans le cadre du projet CARESSE du pôle MSTIC de l'Université Joseph Fourier.

Discussion

Une caractéristique de mon travail de recherche est l'utilisation de concepts et de techniques se trouvant à l'interface de plusieurs disciplines. Ainsi, la première partie de ce mémoire concerne les notions de simulation et bisimulation approchées qui sont des adaptations d'outils classiques en informatique. De manière à calculer des abstractions symboliques pouvant servir à la synthèse de contrôleur, ces notions sont utilisées en combinaison avec des techniques de Lyapunov, qui sont très répandues en automatique. Mon travail sur l'analyse d'atteignabilité utilise principalement des techniques des mathématiques appliquées (approximations numériques, géométrie computationnelle, analyse convexe...), cependant, toute l'approche a été inspirée par le domaine de l'informatique appelé "model checking". Enfin, le domaine entier des systèmes dynamiques multi-agents a été développé à l'interface de la théorie des systèmes dynamiques et de la théorie des graphes. Il y a, à mon avis, un potentiel énorme pour le développement d'approches innovantes à l'interface des mathématiques appliquées, de l'automatique et de l'informatique.

Une autre caractéristique de mon travail est l'intérêt constant porté aux algorithmes et au calcul. La plupart de mes contributions théoriques ont été motivées par le développement

d'approches computationnelles pour l'analyse et le contrôle des systèmes hybrides. Ainsi, mon travail sur les abstractions approximativement bisimilaires a été concrétisé dans l'outil pour la synthèse de contrôleur CoSyMA; les algorithmes pour l'analyse d'atteignabilité basés sur les fonctions support constituent le coeur de la plate-forme de vérification des systèmes hybrides SpaceEx; mon travail sur la dynamique d'opinions a été motivé par le développement d'algorithmes distribués efficaces pour la détection de communautés dans des grand réseaux... Je suis convaincu qu'au regard de la complexité croissante des systèmes dynamiques que l'on cherche à analyser ou contrôler, les techniques computationnelles, possiblement associées à des approches analytiques, deviendront des outils indispensables.

Contents

Chapter 1

Introduction

Hybrid systems are dynamical systems exhibiting both continuous and discrete behaviors. Motivated by the multiplication of "discrete" embedded computing devices interacting with the "continuous" physical world (an application domain that is today referred to as cyber-physical systems), the research on hybrid systems has rapidly developed since the nineties at the intersection of computer science, control theory and applied mathematics. Each discipline has brought its own models and methods and their combination has allowed the scientific community to build the foundations of a theory of hybrid systems. The notion of hybrid automaton [Hen96, LJS+03], which is the most commonly used mathematical model of hybrid systems, combines differential equations and finite state automata and is a typical example of this cross-fertilization. More generally, hybrid systems research has enabled the development of new approaches in computation and control that would not have been possible without the tight interactions between disciplines. For instance, the area of *symbolic control* [EFE06, Tab09] has borrowed from computer science tools and concepts such as formal verification, reachability analysis, abstraction or logics, and has used these in control systems design. In symbolic control, continuous behaviors are abstracted over a finite set of symbols, each symbol representing infinitely many states. These approaches allow one to address specifications that are often different from traditional properties in control theory (e.g. stability, controllability, observability...): such specifications can for instance be given by some logic formula describing the acceptable temporal behaviors of the system. Nevertheless, successful approaches often borrow as much from computer science as they do from control theory and applied mathematics (use of Lyapunov functions, numerical approximations...). Finally, symbolic control also gives a predominant place to algorithms and to the development of computational techniques for analysis and control of dynamical systems.

Most of my own research falls within these domains of hybrid systems and symbolic control, with a focus on the development of computational techniques. I have also worked on the analysis of multi-agent dynamical systems. This document presents the main contributions of my research work since my Ph.D. in 2004. The presentation is not exhaustive but focuses on the results that I consider as the most significant. I also highlight a certain number of results that have been obtained in collaboration with young researchers, Ph.D. students and postdoctoral researchers, that I have supervised. A complete list a my publications is given in appendix.

1.1 Main Contributions

This document is organized in three main parts. The first part introduces an approximation framework that applies to continuous, discrete and hybrid dynamical systems (Chapter 2); several applications of this framework are then presented (Chapters 3 and 4). The second part focuses on reachability analysis (Chapter 5), a valuable computational technique for analysis and control of hybrid systems. Finally, the third part deals with multi-agent dynamical systems (Chapter 6). In the following, we give a brief description of each part.

1.1.1 Approximate simulation and bisimulation

Theories of approximation are fundamental for the analysis and control of complex dynamical systems. For continuous dynamical systems, approximation is traditionally characterized through metrics measuring the distance between the behaviors of two systems (see e.g. [ASG00]). For discrete systems, where a natural notion of distance between behaviors may not exist, approximation is generally tackled through order or equivalence relationships such as language inclusion, simulation and bisimulation relations [Mil89, CGP00]. A major challenge of hybrid system theory is to propose a common framework for approximation of continuous, discrete and hybrid dynamics. Several works have for instance extended simulation and bisimulation relationships to continuous and hybrid systems [Pap03, vdS04, HTP05]. In [GP07b], we have introduced the notions of approximate language inclusion, approximate simulation and bisimulation relations and defined an associated hierarchy of approximation metrics for (continuous, discrete or hybrid) systems observed over metric spaces[1]. Intuitively, these metrics measure how well one system is approximated by an other based on the distance between their observed behaviors; the metrics being equal to zero coinciding with the notions of "exact" language inclusion, simulation and bisimulation. We established a functional characterization of approximate simulation and bisimulation using simulation and bisimulation functions that can be defined by Lyapunov like inequalities.

Approximate bisimulation has shown to be a powerful tool for abstraction based control synthesis. In [Gir12], we have presented approaches for synthesizing controllers for safety and reachability specifications using approximately bisimilar abstractions. Given a controller for an abstraction, we can derive a controller for the original system using specific concretization procedures. The approximate bisimulation relationship between the system and its abstraction allows us to ensure that this controller is "correct by design" meaning that the specification is met by the original system. Moreover, the performances of the maximal (for safety) or time-optimal (for reachability) controllers can be approached arbitrarily close by using abstractions that are precise enough. These approaches can be used to synthesize controllers for classes of incrementally stable (continuous or hybrid) dynamical systems for which we have shown that approximately bisimilar abstractions of arbitrary precision can be computed [Gir07, PGT08, GPT10]. Since these abstractions are symbolic, it allows us to leverage techniques developed in the areas of discrete systems to control continuous or hybrid systems. We have proposed techniques for reducing the complexity of the resulting symbolic controllers, using state quantization and an ef-

[1] This paper was awarded the George S. Axelby Outstanding Paper Award of the IEEE Control System Society in 2009.

ficient representation of the control map [Gir13]. We have also developed an approach for reducing the algorithmic cost of controller synthesis by using multi-scale symbolic abstractions [CGG11b, CGG11a]. Synthesis algorithms exploiting the specificities of these abstractions have been implemented in the tool CoSyMA [MGG13]: the abstractions are computed on the fly during controller synthesis and the dynamics at the finest scales are explored only when necessary. Experimental results show a significant reduction of the algorithmic cost of controller synthesis.

We have explored other applications of approximate simulation and bisimulation besides the symbolic control approach mentioned above. For instance, the notion of simulation function has been used to formally relate the behaviors of two continuous control systems in order to design hierarchical controllers [GP09]. We established effective characterizations of approximate simulation and bisimulation relations in order to compute approximations of continuous or hybrid systems [GP07a, GJP08]. Finally, auto-bisimulation functions (bisimulation functions between a system and itself) have been used to design verification algorithms that can check that a given property holds for an infinite number of trajectories by only simulating a finite number of them [FGP06].

1.1.2 Reachability analysis

Reachability analysis has been a major issue in hybrid systems research. This approach, inspired by ideas from algorithmic verification of discrete systems and numerical simulation of continuous systems, seeks to compute (an approximation of) the set of all trajectories of a system, for all admissible values of initial states and parameters, and under all possible disturbances. A single successful reachability analysis can thus replace infinitely many simulations of individual trajectories [Mal11]. In addition, algorithms for reachability analysis of continuous systems form the core of several computational approaches to solve hybrid systems problems such as controller synthesis, verification or computation of symbolic abstractions [ABD+00, TMBO03, ADI06]. For these approaches to be mathematically sound, we often need to guarantee some properties of the computed approximation. For instance, it is often necessary that the computed approximation includes the true reachable set. Thus, my contributions to that domain deal primarily with algorithms for computing over-approximations of the set of states that are reachable by a continuous dynamical system.

For linear systems with bounded input values, we have proposed time-discretization schemes allowing us to compute an over-approximation of the set of reachable states on a bounded time interval [Gir05, LG10]. This approximation consists in a finite union of compact convex sets and can be made arbitrarily accurate by reducing the time step. We have developed several implementations of these approximation schemes. The first one is based on zonotopes [Gir05], a class of polytopes with interesting computational features. A determinant improvement of this approach, based on the introduction of auxiliary sequences of sets, was brought in [GLM06], opening the way to the computation of very accurate approximations, even in high-dimensions. This technique has been extended to work with arbitrary compact convex sets by proposing an implementation based on support functions [LG10]. The adaptation of these algorithms for reachability analysis of hybrid automata has been described in [GL08, LG09]. It is noticeable that the support function algorithm constitutes the core of the state of the art hybrid system verification platform SpaceEx [FLD+11] developed at Verimag.

We have also worked on reachability analysis of nonlinear systems. For discrete-time polynomial systems, we have developed an approach for computing polytopic over-approximations of the reachable set using relaxations of polynomial optimization problems given by linear programs obtained thanks to the properties of Bernstein polynomials [BTDG12]. For general nonlinear systems, reachability analysis can tackled via the hybridization principle [ADG03, ADG07], which consists in approximating a nonlinear dynamics by a piecewise linear dynamics where additional bounded disturbances are added to account for the approximation error. The reachable set of the resulting hybrid automaton, computable with approaches mentioned above, provides an over-approximation of the reachable set of the original non-linear system.

1.1.3 Multi-agent dynamical systems

The central issue of multi-agent dynamical systems consists in understanding how coordinated global behaviors can emerge from simple local interaction rules between agents. The most studied problem in this area is the consensus problem where the agents seek to reach an agreement in a distributed manner. Consensus algorithms have several applications as different as multi-vehicle control in robotics [RBA07] or modeling of opinion dynamics in social sciences [HK02]. Sufficient conditions for convergence to a consensus are typically based on the topology of the network describing the interactions between agents and on the strength of these interactions. In [MG13], we have established a set of sufficient conditions for achieving consensus in continuous-time. These essentially states that the connectivity in the interaction network should be persistent and that the divergence between reciprocal interactions weights should not be too fast. We have also provided an estimate of the convergence rate to the consensus and shown that our result generalizes several conditions existing in the literature, including those established recently in [HT12].

We have also proposed and studied a model of opinion dynamics that can explain the formation of communities inside a network of agents [MG11]. In our model, agents seek to reach a consensus no slower than a certain convergence rate. It can be seen as a negotiation process where an agent expects that others move sufficiently fast towards its opinion in order to keep negotiating. Then, consensus may not be achieved globally but only in some subgroups of agents that we call communities. We have established a characterization of these communities in terms of algebraic properties of the interaction graph. Moreover, we could check experimentally, on a certain number of benchmarks, that these communities coincides with those observed in reality. Hence, our model provides a naturally distributed solution to the community detection problem in complex networks.

1.2 Research Supervision and Coordination

Since 2004, I have had the honor to supervise the work of several young researchers, Ph.D. students and postdoctoral researchers. In the following, I give a very brief description of these fruitful collaborations and give pointers to publications[2] and to sections of this document where those works are presented. I have also coordinated several research projects which are listed at the end of the section.

[2]The publication numbers refer to the list of publications given in appendix

1.2.1 Ph.D. students

Colas Le Guernic (co-supervised with O. Maler), 2006-2009.
Thesis: Reachability analysis of hybrid systems with linear continuous dynamics.
Publications: J10, C27, C24, C23, C15, C10, CL2, CL1.
Current situation: Researcher, DGA-MI, Rennes.

This thesis deals with the development of accurate and scalable algorithms for reachability analysis of a class of hybrid systems where the continuous dynamics is described by linear differential equations. The main contributions of this thesis are a determinant improvement of the algorithm for the computation of the reachable set of linear systems, its implementations based on zonotopes and support functions and its extension to reachability analysis of hybrid systems. Part of this work is presented in Section 5.1.

Samuel Martin (co-supervised with G. James), 2009-2012.
Thesis: Coordination and robustness of multi-agent dynamical systems.
Publications: J20, J14, C37, C33, C25.
Current situation: Assistant professor (Maître de conférences), Université de Lorraine.

This thesis deals with the analysis of multi-agent dynamical systems. The main contribution of this thesis is a fine analysis of the linear consensus in continuous-time which enables to propose a set of sufficient conditions that are more general than those in the literature. Another important contribution is the proposition of a measure for estimating the robustness of multi-agent formations with respect to the flocking behavior. Part of this work is presented in Section 6.1.

Mohamed Amin Ben Sassi (co-supervised with G. James), 2009-2013.
Thesis: Analysis and control of polynomial dynamical systems.
Publications: J18, J13, C43, C40.
Current situation: Postdoctoral researcher, Université de Grenoble.

This thesis deals with the development of algorithms for analysis and control of polynomial dynamical systems using linear programming. New linear programming relaxations of polynomial optimization problems are obtained either through the blossoming principle or properties of the Bernstein polynomials. The main contributions are algorithms based on these relaxations for reachability analysis, invariant computation and robust control of polynomial systems. Part of this work is presented in Section 5.2.

Pierre-Olivier Lamare (co-supervised with C. Prieur), since 2012.
Thesis: Modeling, simulation and control of switched hyperbolic systems.
Publications: C46.

This thesis deals with switching control of hyperbolic partial differential equations. Based on local measurements and using Lyapunov techniques, a set of stabilizing switching control laws are derived. Issues such as existence of the closed loop solutions, numerical simulation, computational techniques for controller synthesis and applications to control of physical networks are also considered. This work in progress is not presented in this document.

Pierre-Jean Meyer (co-supervised with E. Witrant), since 2012.
Thesis: Hybrid control for green buildings.
Publications: C49.

The goal of this thesis is to apply the symbolic control approach in order to manage energy in buildings. The thermal dynamics of a building is modeled by a hybrid system with disturbances. Symbolic abstractions are computed using the monotonicity properties of the model. Strategies for energy management are then synthesized using the symbolic abstractions, based on several assumptions on the nature (adversarial or stochastic) of disturbances. This work in progress is not presented in this document.

Ying Tang (co-supervised with C. Prieur), since 2012.
Thesis: Analysis of singularly perturbed hyperbolic systems.
Publications: C45, C47.

This thesis deals with the analysis of hyperbolic partial differential equations with multiple time scales. Following a singular perturbation approach, we define the boundary-layer and reduced systems. Using Lyapunov techniques, conditions are established showing the validity of the approximation given by the reduced system. An application to a Poiseuille flow is considered. This work in progress is not presented in this document.

1.2.2 Postdoctoral researchers

Gang Zheng , 2008.
Subject: Verification algorithms based on auto-bisimulation functions.
Publications: J17, C26.
Current situation: Researcher (Chargé de recherche), INRIA, Lille.

We have developed verification algorithms for dynamical systems with inputs that can determine whether a given safety or bounded liveness property is satisfied by the infinitely many trajectories of the system. The algorithm only needs to compute a finite number of trajectories and an estimate of how robustly these satisfy the specified property. Then, one can infer that the property is satisfied by all trajectories in neighborhoods that are characterized by an auto-bisimulation function. Part of this work is presented in Section 4.3.

Constantin Morarescu , 2009.
Subject: Consensus and community detection in networks.
Publications: J11, C31, C30, C29.
Current situation: Assistant professor (Maître de conférences), Université de Lorraine.

We have proposed and studied a model of opinion dynamics with decaying confidence. In this model, global consensus may not be achieved and subgroups of agents may only agree locally, organizing themselves in communities. We have established an algebraic characterization of these communities. Our model can serve as an elegant solution for a distributed algorithm for detecting communities in large networks. This work is presented in Section 6.2.

Javier Camara (co-supervised with G. Goessler), 2010.
Subject: Multi-scale symbolic abstractions of switched systems.
Publications: C34, C36.
Current situation: Postdoctoral researcher, Carnegie Mellon University, USA.

We have established the existence of multi-scale symbolic abstractions for a class of incrementally stable systems. We have studied the use of these multi-scale symbolic abstractions for controller synthesis. For safety specifications, we have proposed the notion of maximal lazy safety controller and developed a multi-scale algorithm for its computation. This work is presented in Section 3.3.

Sebti Mouelhi (co-supervised with G. Goessler), 2011-2012.
Subject: Multi-scale symbolic abstractions of switched systems.
Publications: C42.
Current situation: Research engineer, Safe River, Paris.

We have extended the algorithm for synthesizing the maximal lazy safety controller in order to compute controllers for bounded-time reachability specifications. These algorithms have been implemented in the tool CoSyMA. This work is presented in Section 3.3.

Euriell Le Corronc (co-supervised with G. Goessler), 2012-2013.
Subject : Grid-free symbolic abstractions of switched systems.
Publications: C48.
Current situation: Assistant professor (Maître de Conférences), Université de Toulouse.

We have explored a new approach for computing approximately bisimilar symbolic abstractions of incrementally stable systems. Contrarily to existing approaches that use a discretization of the continuous state space as the set of symbolic states, we propose to use input sequences of given length as symbolic states. A result showing the existence of an approximate bisimulation relation has been established. This work in progress is not presented in this document.

1.2.3 Research projects

Most of my research work has been carried out within funded research projects, most of which I have coordinated. These projects are listed below.

VAL-AMS: High Confidence Validation of Analog and Mixed Signal Circuits.
Type: ANR-SETIN (2007-2009)
Partners: Verimag, Laboratoire Jean Kuntzmann, INRIA.
Personal role: Scientific responsible for LJK.

CARESSE: Contrôle et Analyse de Réseaux de Systèmes Dynamiques Évolutifs.
Type: UJF-MSTIC (2008-2009)
Partners: Laboratoire Jean Kuntzmann
Personal role: Principal investigator.

VEDECY: Verification and Design of Cyber-Physical Systems.
Type: ANR-ARPEGE (2009-2012)
Partners : Laboratoire Jean Kuntzmann, Verimag, INRIA.
Personal role: Principal investigator.

SYMBAD: Symbolic Approaches to the Design of Cyber-Physical Systems.
Type: UJF-MSTIC (2012-2013)
Partners: Laboratoire Jean Kuntzmann, INRIA.
Personal role: Principal investigator.

COHYBA: Contrôle Hybride pour les Bâtiments Verts.
Type: CIBLE, Région Rhône Alpes (2012-2014)
Partners: Laboratoire Jean Kuntzmann, GIPSA Lab.
Personal role: Principal investigator.

COMPACS: Computation Aware Control Systems.
Type: ANR-Blanc (2013-2017)
Partners: Laboratoire Jean Kuntzmann, CRAN, Verimag.
Personal role: Principal investigator.

Part I

Approximate Simulation and Bisimulation

Chapter 2

Approximation Framework for Discrete and Continuous Systems

Résumé : *Un défi essentiel du domaine des systèmes hybrides est de pouvoir considérer des dynamiques continues et discrètes dans un cadre unique. Les théories d'approximation des systèmes, qui sont fondamentales pour l'analyse et la synthèse de systèmes complexes, ont été développées de manières indépendantes pour les systèmes dynamiques discrets et continus. Dans ce chapitre, nous présentons un cadre théorique d'approximation qui s'applique aux deux types de systèmes. Ce cadre, introduit dans [GP07b], est fondé sur une hiérarchie de métriques d'approximation généralisant la hiérarchie usuelle des relations d'inclusion de langage, simulation et bisimulation qui constituent les outils usuels pour l'approximation des systèmes dynamiques discrets. Intuitivement, ces métriques mesurent la qualité de l'approximation d'un système par un autre en considérant la distance entre leurs comportements observés. Les relations traditionnelles sont capturées lorsque ces métriques sont mises à* 0. *Les notions centrales de notre approche sont les relations de simulation ou de bisimulation approchée et leur caractérisation fonctionnelles appelées fonctions de simulation et bisimulation et définies par inégalités de type Lyapunov. En particulier, ces fonctions se montrent très utiles pour calculer des bornes garanties de nos métriques d'approximation.*

A major challenge in the area of hybrid systems is to think about continuous and discrete dynamics in a unified systems theoretic foundation. In particular, theories of system approximation, which are crucial for the application of analysis and synthesis techniques to complex systems, have been developed independently on both sides. For continuous systems, approximation is traditionally tackled through metrics, for instance between transfer functions [ASG00] as in the established domain of model reduction. For discrete systems, approximation is usually specified from the set theoretic point of view: usual abstraction relationships such as language inclusion, simulation and bisimulation relations [Mil89, CGP00] require inclusion or equality of the systems observed behaviors. In the past decade, these notions have been extended to continuous and hybrid systems [Pap03, vdS04, HTP05] providing a basis for a common theory of system approximation.

The notions of language inclusion, simulation, and bisimulation for both discrete and continuous systems are all "exact", requiring outputs of two systems to match exactly. As these exact relationships between discrete systems do not permit any error, there are clear limitations in the amount of system compression or approximation that can be achieved. Approximate relationships which do allow for the possibility of a quantifiable error, certainly allows for more dramatic system compression. This has been the tradition for continuous systems, and it has also been argued [CB02, dAFS04] that in several cases, quantitative notions for discrete or hybrid system approximation are not only better candidates for complexity reduction but also provide more robust relationships between systems.

In this chapter, we review a framework for system approximation that applies to both discrete and continuous transition systems by providing quantitative generalizations of language inclusion, simulation, and bisimulation. Our approximation framework has been introduced in [GP07b] and applies equally to discrete and continuous systems. It is based on a hierarchy of approximation metrics, which generalizes the usual relationship hierarchy of language inclusion, simulation and bisimulation. These metrics essentially quantify how well a system is approximated by another based on the distance between their observed behaviors. The traditional relationships are captured as the zero sections of these approximation metrics. The central notions in this framework are that of approximate simulation and bisimulation relations and their functional characterizations called simulation and bisimulation functions and defined by Lyapunov-type inequalities. In particular, these functions show to be very useful to compute guaranteed upper-bounds on the approximation metrics.

2.1 Hierarchy of Approximation Metrics

In this section, we introduce a theoretical approximation framework based on a hierarchy of metrics generalizing the notions of language inclusion, simulation, and bisimulation.

2.1.1 Transition systems

We consider transition systems which enables us to model in a common framework discrete, continuous and hybrid systems with either deterministic or non-deterministic dynamics (see e.g. [AHLP00, Tab09]).

Definition 2.1 (Transition systems) *A transition system $T = (X, U, \mathcal{S}, X^0, Y, \mathcal{O})$ consists of a set of states X; a set of inputs U; a transition map $\mathcal{S} : X \times U \to 2^X$; a set of initial states $X^0 \subseteq X$; a set of outputs Y; and an output map $\mathcal{O} : X \to Y$.*

T is *metric* if the sets of states X and outputs Y are equipped with metrics. If the set of states X and inputs U are countable or finite, then T is said *discrete* or *symbolic*, respectively. The transition map captures the dynamics of the system: $x' \in \mathcal{S}(x, u)$ means that the state of the system can evolve from x to x' under the action of input u. Input $u \in U$ belongs to the set of *enabled inputs* at state $x \in X$, denoted $\mathsf{Enab}(x)$, if $\mathcal{S}(x, u) \neq \emptyset$. T is said to be *deterministic* if for all $x \in X$, for all $u \in \mathsf{Enab}(x)$, $\mathcal{S}(x, u)$ consists of a unique element. State $x \in X$ is said to be *blocking* if $\mathsf{Enab}(x) = \emptyset$, otherwise it is said to be *non-blocking*. T is said to be non-blocking if all states are non-blocking. It is sometimes

necessary to require some additional technical assumptions on metric transition systems, such as local compactness of the set of states X; continuity of the map $\mathcal{O}$ and of the set-valued map $\mathcal{S}$; compactness of the set of initial states X^0 and of $\mathcal{S}(x,u)$ for all $x \in X$, $u \in U$; and open support of the set-valued map $\mathcal{S}(.,u)$, for all $u \in U$. Metric transition systems satisfying these assumptions are called *regular*.

A *state trajectory* is a sequence of states and inputs of the form $\sigma_X = (x^0,u^0),(x^1,u^1),\ldots,(x^{N-1},u^{N-1}),x^N$, where $x^0 \in X^0$ and for all $i = 0,\ldots,N-1$, $x^{i+1} \in \mathcal{S}(x^i,u^i)$; the associated *output trajectory* is a sequence of outputs and inputs $\sigma_Y = (y^0,u^0),(y^1,u^1),\ldots,(y^{N-1},u^{N-1}),y^N$, where for all $i = 0,\ldots,N$, $y^i = \mathcal{O}(x^i)$. $\mathsf{l}(\sigma_X) = \mathsf{l}(\sigma_Y) = N$ is called the *length* of σ_X and σ_Y; it is also possible to consider infinite sequences, in that case $\mathsf{l}(\sigma_X) = \mathsf{l}(\sigma_Y) = +\infty$. The set of output trajectories of T, denoted $\mathcal{L}(T)$, is called the *observed behavior* or the *language* of transition system T. A state $x \in X$ is said to be *reachable* if there exists a state trajectory $\sigma_X = (x^0,u^0),(x^1,u^1),\ldots,(x^{N-1},u^{N-1}),x^N$ with $x^N = x$.

Example 2.1 *Let us show how transition systems can serve to describe the dynamics of a continuous system given by*

$$\Sigma: \begin{cases} \dot{\mathbf{x}}(t) = f(\mathbf{x}(t)), & \mathbf{x}(0) \in I,\ \mathbf{x}(t) \in \mathbb{R}^n, \\ \mathbf{y}(t) = g(\mathbf{x}(t)), & \mathbf{y}(t) \in \mathbb{R}^p \end{cases}$$

Following [Pap03], we define the associated transition system $T(\Sigma) = (X,U,\mathcal{S},X^0,Y,\mathcal{O})$ where the set of states is $X = \mathbb{R}^n$; the inputs stand for the time $U = \mathbb{R}^+$; the set of initial states is $X^0 = I$; the set of outputs is $Y = \mathbb{R}^p$; the output map is $\mathcal{O} = g$; and the transition relation is given by the flow of the differential equation: $x' \in \mathcal{S}(x,t)$ if and only if there exists a differentiable function $\mathbf{x}$ such that $\mathbf{x}(0) = x$, $\mathbf{x}(t) = x'$ and for all $s \in [0,t]$, $\dot{\mathbf{x}}(s) = f(\mathbf{x}(s))$. Let us assume that I is compact, g is continuous and f is locally Lipschitz continuous, then we can show that $T(\Sigma)$ is a deterministic, non-blocking, regular metric transition system when the set of states and observations are equipped with the Euclidean distance. We shall see other descriptions of continuous or hybrid dynamics by means of transition systems in the following chapters.

2.1.2 Behavioral pseudo-metrics

We want to quantify the distance between the behaviors of two metric transition systems, possibly of different nature (e.g. one can be continuous and the other one discrete or symbolic). Let us consider two metric transition systems $T_j = (X_j,U,\mathcal{S}_j,X_j^0,Y,\mathcal{O}_j)$, $j \in \{1,2\}$, with common sets of inputs and outputs and let $\sigma_j = (y_j^0,u_j^0),(y_j^1,u_j^1),\ldots$ be output trajectories of these systems. The set of outputs Y is equipped with a metric d. The distance between the output trajectories σ_1 and σ_2 is defined as

$$d_\infty(\sigma_1,\sigma_2) = \begin{cases} \sup\limits_{i \le \mathsf{l}(\sigma_1)} d(y_1^i,y_2^i) & \text{if } \mathsf{l}(\sigma_1) = \mathsf{l}(\sigma_2) \text{ and } \forall i = 0,\ldots,\mathsf{l}(\sigma_1)-1, u_1^i = u_2^i; \\ +\infty & \text{otherwise.} \end{cases}$$

Essentially, the distance between σ_1 and σ_2 is finite if they have the same length and the same sequence of inputs; in that case the distance between the trajectories is the maximal distance between the sequences of outputs.

We can now define metrics measuring the distance between the languages of T_1 and T_2 and generalizing the notion of language inclusion:

Definition 2.2 (Language metrics) *The directed and undirected language metrics are defined respectively as*

$$\begin{aligned} d_L^{\rightarrow}(T_1, T_2) &= \sup_{\sigma_1 \in \mathcal{L}(T_1)} \inf_{\sigma_2 \in \mathcal{L}(T_2)} d_\infty(\sigma_1, \sigma_2) \\ d_L(T_1, T_2) &= \max\left\{d_L^{\rightarrow}(T_1, T_2), d_L^{\rightarrow}(T_2, T_1)\right\}. \end{aligned}$$

The meaning of the directed language metric is as follows: for any output trajectory of the system T_1, one can find an output trajectory of the system T_2, with the same sequence of inputs, such that the distance between the sequence of outputs of the two systems remains bounded by $d_L^{\rightarrow}(T_1, T_2)$. In addition, if $\mathcal{L}(T_1) \subseteq \mathcal{L}(T_2)$ then $d_L^{\rightarrow}(T_1, T_2) = 0$. One can show that the language metrics are actually directed and undirected pseudo-metrics[1] on the set of metric transition systems.

The computation of $d_L^{\rightarrow}(T_1, T_2)$ and $d_L(T_1, T_2)$ is generally extremely difficult, particularly for non-deterministic systems. Though, we can define a hierarchy of stronger metrics, that are easier to compute and based on approximate versions of the notions of simulation and bisimulation relations [Mil89, CGP00]. The notion of "exact" simulation relation has been traditionally used in computer science as a mean of abstraction of transition systems. Essentially, a simulation relation of T_1 by T_2 is a relation on the states of the systems that describes how to select transitions of T_2 in order to match the transitions of T_1 and to produce the same output sequence than T_1. The notion of approximate simulation relation is obtained by relaxing the equality of outputs: instead of requiring them to be identical, we require that they remain within some specified distance.

Definition 2.3 (Approximate simulation) *Let $\varepsilon \geq 0$, a relation $\mathcal{R}_\varepsilon \subseteq X_1 \times X_2$ is called an approximate simulation relation of T_1 by T_2, of precision ε, if for all $(x_1, x_2) \in \mathcal{R}_\varepsilon$:*

1. *$d\left(\mathcal{O}_1(x_1), \mathcal{O}_2(x_2)\right) \leq \varepsilon$,*
2. *$\forall u \in U$, $\forall x_1' \in \mathcal{S}_1(x_1, u)$, $\exists x_2' \in \mathcal{S}_2(x_2, u)$ such that $(x_1', x_2') \in \mathcal{R}_\varepsilon$.*

T_2 approximately simulates T_1 with precision ε (denoted $T_1 \preceq_\varepsilon T_2$), if there exists $\mathcal{R}_\varepsilon$, an approximate simulation relation of T_1 by T_2, of precision ε, such that for all $x_1 \in X_1^0$, there exists $x_2 \in X_2^0$ such that $(x_1, x_2) \in \mathcal{R}_\varepsilon$.

For $\varepsilon = 0$, we recover the established definition of exact simulation relation (denoted $T_1 \preceq T_2$). Approximate bisimulation is defined in a similar way as the symmetric version of approximate simulation:

Definition 2.4 (Approximate bisimulation) *Let $\varepsilon \geq 0$, a relation $\mathcal{R}_\varepsilon \subseteq X_1 \times X_2$ is called an approximate bisimulation relation between T_1 and T_2, of precision ε, if for all $(x_1, x_2) \in \mathcal{R}_\varepsilon$:*

1. *$d\left(\mathcal{O}_1(x_1), \mathcal{O}_2(x_2)\right) \leq \varepsilon$,*

[1] A metric on a set E is a positive function $d : E \times E \to \mathbb{R} \cup \{+\infty\}$, such that the three following properties hold: for all $e_1 \in E$, $e_2 \in E$, $e_3 \in E$, $d(e_1, e_3) \leq d(e_1, e_2) + d(e_2, e_3)$; for all $e_1 \in E$, $e_2 \in E$, $d(e_1, e_2) = 0 \iff e_1 = e_2$; for all $e_1 \in E$, $e_2 \in E$, $d(e_1, e_2) = d(e_2, e_1)$. We say that (E, d) is a metric space. If the second property is replaced by $e_1 = e_2 \implies d(e_1, e_2) = 0$ then d is called a pseudo-metric. If the third property is dropped, then d is called a directed metric.

2. $\forall u \in U$, $\forall x_1' \in \mathcal{S}_1(x_1, u)$, $\exists x_2' \in \mathcal{S}_2(x_2, u)$ *such that* $(x_1', x_2') \in \mathcal{R}_\varepsilon$.

3. $\forall u \in U$, $\forall x_2' \in \mathcal{S}_2(x_2, u)$, $\exists x_1' \in \mathcal{S}_1(x_1, u)$ *such that* $(x_1', x_2') \in \mathcal{R}_\varepsilon$.

T_1 and T_2 are approximately bisimilar with precision ε (denoted $T_1 \sim_\varepsilon T_2$), if there exists $\mathcal{R}_\varepsilon$, an approximate bisimulation relation between T_1 and T_2, of precision ε, such that for all $x_1 \in X_1^0$, there exists $x_2 \in X_2^0$ such that $(x_1, x_2) \in \mathcal{R}_\varepsilon$, and conversely.

Again, for $\varepsilon = 0$, we recover the notion of exact bisimulation relation (denoted $T_1 \sim T_2$). Based on the notions of approximate simulation and bisimulations, we can define metrics that intuitively measures how far two transition systems are from exact simulation or bisimulation.

Definition 2.5 (Simulation and bisimulation metrics) *The simulation and bisimulation metrics are defined respectively by*

$$d_S^{\rightarrow}(T_1, T_2) = \inf \{\varepsilon |\ T_1 \preceq_\varepsilon T_2\},$$

$$d_B(T_1, T_2) = \inf \{\varepsilon |\ T_1 \sim_\varepsilon T_2\}.$$

One can show that the simulation and bisimulation metrics are respectively directed and undirected pseudo-metrics over the set of metric transition systems. Interestingly, the zero sections of these metrics capture the traditional system relationships.

Proposition 2.1 *If T_1 and T_2 are regular metric transition systems then*

$$T_1 \preceq T_2 \iff d_S^{\rightarrow}(T_1, T_2) = 0,$$

$$T_1 \sim T_2 \iff d_B(T_1, T_2) = 0.$$

The main result of the section is the following that establishes a hierarchy between language, simulation and bisimulation metrics:

Theorem 2.1 (Hierarchy of approximation metrics) *For all metric transitions systems T_1 and T_2 with the same sets of inputs and outputs, the following inequalities hold:*

$$\begin{array}{ccc} d_B(T_1, T_2) & \geq & d_L(T_1, T_2) \\ \vee| & & \vee| \\ d_S^{\rightarrow}(T_1, T_2) & \geq & d_L^{\rightarrow}(T_1, T_2) \end{array}$$

In addition, if T_1 and T_2 are deterministic, then

$$d_B(T_1, T_2) = d_L(T_1, T_2) \text{ and } d_S^{\rightarrow}(T_1, T_2) = d_L^{\rightarrow}(T_1, T_2).$$

2.2 Simulation and Bisimulation Functions

In this section, we focus on the computation of the simulation and bisimulation metrics. In the following, we assume that the metric transition systems T_1 and T_2 we consider are regular. We present an approach enabling to compute guaranteed upper-bounds of these metrics, based on the notion of simulation and bisimulation functions defined by Lyapunov like inequalities. Essentially, a simulation function of T_1 by T_2 is a positive function defined on $X_1 \times X_2$, bounding the distance between the outputs associated to the couple (x_1, x_2) and non-increasing under the dynamics of the systems.

Definition 2.6 (Simulation function) *A function $\mathcal{V} : X_1 \times X_2 \to \mathbb{R}^+ \cup \{+\infty\}$ is called a simulation function of T_1 by T_2 if its sub-level sets are closed, and for all $(x_1, x_2) \in X_1 \times X_2$:*

$$\mathcal{V}(x_1, x_2) \geq \max \left\{ d(\mathcal{O}_1(x_1), \mathcal{O}_2(x_2)), \sup_{\substack{u \in U \\ x_1' \in \mathcal{S}_1(x_1, u)}} \inf_{x_2' \in \mathcal{S}_2(x_2, u)} \mathcal{V}(x_1', x_2') \right\}. \tag{2.1}$$

The sub-level sets of a simulation function of T_1 by T_2 provide a convenient way to define approximate simulation relations of T_1 by T_2.

Proposition 2.2 *Let $\mathcal{V}$ be a simulation function of T_1 by T_2. Then, for all $\varepsilon \geq 0$,*

$$\mathcal{R}_\varepsilon = \{(x_1, x_2) \in X_1 \times X_2 |\ \mathcal{V}(x_1, x_2) \leq \varepsilon\}$$

is an approximate simulation relation of T_1 by T_2, of precision ε.

Let us remark that, particularly, the zero set of a simulation function is an exact simulation relation. As a consequence of the previous result, an over-approximation of the simulation metric can be computed using a simulation function.

Proposition 2.3 *Let $\mathcal{V}$ be a simulation function of T_1 by T_2. Then,*

$$d_S^{\rightarrow}(T_1, T_2) \leq \sup_{x_1 \in X_1^0} \inf_{x_2 \in X_2^0} \mathcal{V}(x_1, x_2).$$

Actually, it is possible to show that there exists a particular simulation function satisfying a Bellman equation and for which the upper bound given in the previous theorem is tight.

Theorem 2.2 (Minimal simulation function) *There exists a simulation function of T_1 by T_2, $\mathcal{V}_S^{\min}$ such that for all simulation functions of T_1 by T_2, $\mathcal{V}$, for all $(x_1, x_2) \in X_1 \times X_2$, $\mathcal{V}_S^{\min}(x_1, x_2) \leq \mathcal{V}(x_1, x_2)$. This minimal simulation function is also the smallest solution of the Bellman equation*

$$\mathcal{V}_S^{\min}(x_1, x_2) = \max \left\{ d(\mathcal{O}_1(x_1), \mathcal{O}_2(x_2)), \sup_{\substack{u \in U \\ x_1' \in \mathcal{S}_1(x_1, u)}} \inf_{x_2' \in \mathcal{S}_2(x_2, u)} \mathcal{V}_S^{\min}(x_1', x_2') \right\}. \tag{2.2}$$

Then, the simulation metric can be computed by

$$d_S^{\rightarrow}(T_1, T_2) = \sup_{x_1 \in X_1^0} \inf_{x_2 \in X_2^0} \mathcal{V}_S^{\min}(x_1, x_2).$$

For symbolic transition systems, it is possible to solve the Bellman equation (2.2) and to compute exactly the simulation metrics. However, this is not the case in general, and in practice, we often use the characterization given by Lyapunov like inequalities (2.1) and compute upper-bounds of the simulation metrics. Similarly, the bisimulation metric can be computed or approximated using bisimulation functions, which are essentially symmetric versions of the simulation functions:

Definition 2.7 (Bisimulation function) *A function $\mathcal{V} : X_1 \times X_2 \to \mathbb{R}^+ \cup \{+\infty\}$ is called a bisimulation function between T_1 and T_2 if its sub-level sets are closed, and for all $(x_1, x_2) \in X_1 \times X_2$:*

$$\mathcal{V}(x_1,x_2) \geq \max\left\{ d(\mathcal{O}_1(x_1),\mathcal{O}_2(x_2)), \sup_{\substack{u\in U\\ x_1'\in\mathcal{S}_1(x_1,u)}} \inf_{x_2'\in\mathcal{S}_2(x_2,u)} \mathcal{V}(x_1',x_2'), \right. \tag{2.3}$$
$$\left. \sup_{\substack{u\in U\\ x_2'\in\mathcal{S}_2(x_2,u)}} \inf_{x_1'\in\mathcal{S}_1(x_1,u)} \mathcal{V}(x_1',x_2') \right\}.$$

Results similar to those of simulation functions hold for the case of bisimulation functions. In particular, the sub-level sets of a bisimulation function is an approximate bisimulation relation.

Proposition 2.4 *Let $\mathcal{V}$ be a bisimulation function between T_1 and T_2. Then, for all $\varepsilon \geq 0$,*

$$\mathcal{R}_\varepsilon = \{(x_1,x_2) \in X_1 \times X_2 |\ \mathcal{V}(x_1,x_2) \leq \varepsilon\}$$

is an approximate bisimulation relation between T_1 and T_2, of precision ε.

The zero set of a bisimulation function is an exact bisimulation relation. An over-approximation of the bisimulation metric can be computed using a bisimulation function.

Proposition 2.5 *Let $\mathcal{V}$ be a bisimulation function between T_1 and T_2. Then,*

$$d_B(T_1,T_2) \leq \max\left\{ \sup_{x_1\in X_1^0} \inf_{x_2\in X_2^0} \mathcal{V}(x_1,x_2), \sup_{x_2\in X_2^0} \inf_{x_1\in X_1^0} \mathcal{V}(x_1,x_2) \right\}.$$

Also, there exists a particular bisimulation function allowing us to compute the exact value of the bisimulation metric:

Theorem 2.3 (Minimal bisimulation function) *There exists a bisimulation function between T_1 and T_2, $\mathcal{V}_B^{\min}$ such that for all bisimulation functions between T_1 and T_2, $\mathcal{V}$, for all $(x_1, x_2) \in X_1 \times X_2$, $\mathcal{V}_B^{\min}(x_1,x_2) \leq \mathcal{V}(x_1,x_2)$. This minimal bisimulation function is also the smallest solution of the Bellman equation*

$$\mathcal{V}_B^{\min}(x_1,x_2) = \max\left\{ d(\mathcal{O}_1(x_1),\mathcal{O}_2(x_2)), \sup_{\substack{u\in U\\ x_1'\in\mathcal{S}_1(x_1,u)}} \inf_{x_2'\in\mathcal{S}_2(x_2,u)} \mathcal{V}_B^{\min}(x_1',x_2'), \right. \tag{2.4}$$
$$\left. \sup_{\substack{u\in U\\ x_2'\in\mathcal{S}_2(x_2,u)}} \inf_{x_1'\in\mathcal{S}_1(x_1,u)} \mathcal{V}_B^{\min}(x_1',x_2') \right\}.$$

Then, the bisimulation metric can be computed by

$$d_B(T_1,T_2) = \max\left\{ \sup_{x_1\in X_1^0} \inf_{x_2\in X_2^0} \mathcal{V}_B^{\min}(x_1,x_2), \sup_{x_2\in X_2^0} \inf_{x_1\in X_1^0} \mathcal{V}_B^{\min}(x_1,x_2) \right\}.$$

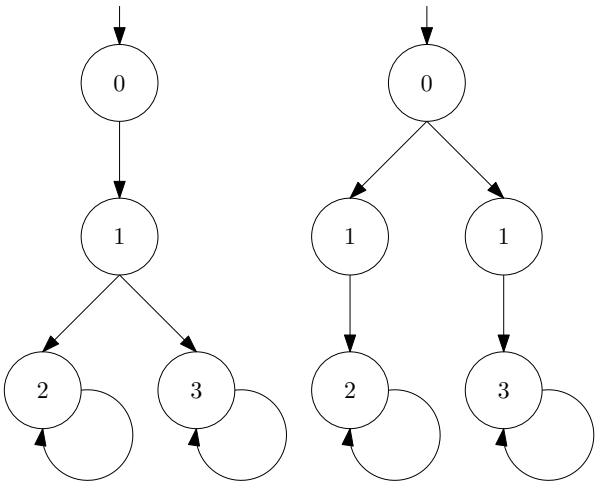

Figure 2.1: These two transition systems generate the same sequences of outputs but are only approximately bisimilar with precision 1.

Example 2.2 *We first consider a simple example for discrete systems. Consider the two symbolic transition systems represented in Figure 2.1, where the common set of outputs is* $\{1, 2, 3, 4\}$*, the common set of inputs is a singleton and thus multiple outgoing transitions from some of the states are manifestations of non-determinism. It is straightforward to verify that both systems can generate the same sequences of outputs:* $0, 1, 2, 2, 2, \ldots$ *and* $0, 1, 3, 3, 3, \ldots$*. Then, it follows that* $d_L(T_1, T_2) = 0$*. The computation of the simulation and bisimulation metrics is also possible by solving the Bellman equations (2.2) and (2.4). Then, we obtain that* $d_S^{\rightarrow}(T_1, T_2) = 1$*,* $d_S^{\rightarrow}(T_2, T_1) = 0$ *and* $d_B(T_1, T_2) = 1$*. Therefore,* T_1 *and* T_2 *are language equivalent though they are not bisimilar but only approximately bisimilar with precision* 1*.*

Example 2.3 *Before considering more involved applications of our approximation framework, we present a simple example showing a procedure for the computation of bisimulation functions for deterministic linear systems. For* $j \in \{1, 2\}$*, let*

$$\Sigma_j : \begin{cases} \dot{\mathbf{x}}_j(t) &= A_j \mathbf{x}_j(t),\ \mathbf{x}_j(t) \in \mathbb{R}^{n_j},\ \mathbf{x}_j(0) \in I_j^0, \\ \mathbf{y}_j(t) &= C_j \mathbf{x}_j(t),\ \mathbf{y}_j(t) \in \mathbb{R}^p. \end{cases}$$

As in Example 2.1, we define associated transition systems $T(\Sigma_j) = (X_j, U, \mathcal{S}_j, X_j^0, Y, \mathcal{O}_j)$ *where the sets of states are* $X_j = \mathbb{R}^{n_j}$*; the common set of inputs is* $U = \mathbb{R}^+$ *and represents time; the common set of outputs is* $Y = \mathbb{R}^p$*; the sets of initial states are* $X_j^0 = I_j^0$*; the observation maps are given by* $\mathcal{O}_j(x_j) = C_j x_j$*; and the transition maps are given by* $x_j' \in \mathcal{S}_j(x_j, t)$ *if and only if* $x_j' = e^{tA_j} x_j$*. The sets of states and outputs are equipped with the usual Euclidean distance, making* $T(\Sigma_j)$ *deterministic, non-blocking, regular metric transition systems. We aim at computing the bisimulation metrics between* $T(\Sigma_1)$ *and* $T(\Sigma_2)$*. For these simple systems, equation (2.4) can be shown to be equivalent to a Hamilton Jacobi Bellman partial differential equation, which is generally hard to solve, particularly for high dimensional systems. Therefore, we shall compute only an upper bound of the bisimulation metrics using a bisimulation function. Restricting our attention*

to functions of the form

$$\mathcal{V}(x_1, x_2) = \sqrt{(x_1, x_2)^\top M (x_1, x_2)},$$

equation (2.3) reduces to the following set of linear matrix inequalities:

$$M \geq C^\top C \text{ and } A^\top M + MA \leq 0$$

where

$$C = [C_1 \quad -C_2] \text{ and } A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}.$$

The first inequality states that the bisimulation function bounds the distance between outputs while the second one ensures that the function decreases during the evolution of the systems. This set of linear matrix inequalities can be efficiently solved using semi-definite programming, even for high dimensional systems. Then, following Proposition 2.5, the obtained bisimulation function can be used to compute an over-approximation of the bisimulation metric. Let us remark that the linear matrix inequalities are always solvable if both systems are stable. Hence, two stable linear systems are always approximately bisimilar, and an upper-bound of the bisimulation metric can always be computed.

Discussion: The results presented in this chapter were developed in collaboration with George J. Pappas during my postdoctoral stay at University of Pennsylvania. All proofs can be found in the paper [GP07b].

Prior to our work, the use of approximation metrics instead of relationships had been considered in the context of probabilistic transition systems [PHW03, vBMOW03, DGJP04] where it is natural to consider approximations of the transition probabilities. The earliest work on a notion of approximate bisimilarity for non-probabilistic transition systems can be found in [YW00] where the notion of *bisimulation index* can be related to that of precision in Definition 2.4. Our work is more related to that of [dAFS04] where the notion of *branching distance* defined in that work is actually close to the notion of minimal bisimulation function, satisfying equation (2.4), between a system and itself. Similar ideas were also explored in [vB05, HMP05]. A construction equivalent to our approximate simulation relations was also introduced in [Tab06] based on the use of set valued output maps.

There has been several extensions of the approximation framework presented in this section, essentially by relaxing the equality of inputs required in the definition of approximate simulation and bisimulation. One approach consists in considering metrics on the set of states and on the set of inputs [JDBP09, QFD11a]; another approach extends the framework by defining the notion of *alternating approximate (bi)simulation* [PT09].

Chapter 3

Controller Synthesis using Approximate Bisimulation

Résumé : *L'utilisation de systèmes symboliques ou discrets comme abstractions de la dynamique continue est devenue une approche commune pour le contrôle des systèmes hybrides. Le bénéfice de cette approche est double : d'abord en abstrayant la dynamique continue, la synthèse de contrôleur peut être résolue par les méthodes algorithmiques existantes pour les systèmes dynamiques discrets. Ensuite, si l'on peut établir une relation formelle entre les comportements du système original et de son abstraction, le contrôleur obtenu est "correct par conception". Ce chapitre traite de l'utilisation d'abstractions approximativement bisimilaires pour la synthèse de contrôleurs pour des spécifications de sûreté ou d'atteignabilité. Dans la première partie, nous présentons des approches spécifiques pour résoudre les deux classes de problèmes [Gir12]. Pour les contrôleurs obtenus nous fournissons des estimations de leur distance au contrôleur maximal (pour la sûreté) ou optimal (pour l'atteignabilité). Dans la deuxième partie, nous appliquons ces approches à la synthèse de contrôleurs pour des systèmes dynamiques à commutation. Nous montrons que pour une classe de systèmes incrémentalement stables, il est toujours possible de calculer des abstractions symboliques approximativement bisimilaires dont la précision peut être arbitrairement choisie [GPT10]. Nos approches nous permettent donc de synthétiser, pour des systèmes dynamiques à commutation, des contrôleurs pour la sûreté ou l'atteignabilité qui sont corrects par conception; de plus, les contrôleurs maximaux et optimaux peuvent être approchés aussi prêt que souhaité. Nous discutons également la réduction de la complexité des contrôleurs, en utilisant des techniques de quantification de l'état et une représentation efficace de la loi de commande [Gir13]. Dans la dernière partie du chapitre, nous présentons une approche pour réduire le coût algorithmique de la synthèse de contrôleurs, basée sur des abstractions symboliques multi-échelles [CGG11b, CGG11a]. Des algorithmes de synthèse qui exploitent les spécificités de ces abstractions ont été implémentés dans l'outil CoSyMA [MGG13] : celles-ci sont calculées à la volée et les échelles les plus fines de l'abstraction ne sont explorées que lorsque cela est nécessaire. Des résultats expérimentaux montrent une amélioration significative de la complexité de la synthèse de contrôleurs.*

The use of symbolic or discrete models as abstractions of the continuous dynamics has become a standard approach to hybrid systems design (see for instance [RO98, MR99, HvS01, TP06, KB06, Rei09]). The benefit of this approach is double. Firstly, by abstracting the continuous dynamics, controller synthesis problems can be efficiently solved using techniques developed in the areas of supervisory control of discrete-event systems or algorithmic game theory. Secondly, if the behaviors of the original system and of the discrete abstraction are formally related by some (exact or approximate) behavioral relationship, the synthesized controller can be shown to be "correct by design" and thus the need of formal verification is reduced.

This chapter deals with the synthesis of controllers using approximately bisimilar abstractions with an emphasis on safety and reachability problems. Safety problems consist in synthesizing a controller that restricts the behaviors of a system so that its outputs remain in some specified safe set. One is usually interested in designing a controller that is as permissive as possible since this makes it possible, using modular approaches, to ensure, a posteriori, secondary control objectives (see e.g. [RW87]). Reachability problems consist in synthesizing a controller that steers the observations of the system to some target region while keeping them in a given safe set along the way. In addition, in order to choose among the possible controllers, we try to minimize the time to reach the target. Hence, we consider a time-optimal control problem.

In the first part of this chapter, we propose abstraction-based approaches to solve both classes of problems. We start by synthesizing a controller for an approximately bisimilar abstraction of our original system. Then, using specific concretization procedures, we obtain a controller for our original system that is proved "correct by design". For safety problems, we provide estimates of the distance between the synthesized controller and the maximal (i.e the most permissive) safety controller. For reachability problems, we provide estimates of the distance between the performances of the synthesized controller and of the time-optimal controller.

In the second part of this chapter, we apply these approaches to a class of switched systems. Switched systems constitute an important modeling paradigm faithfully describing many engineering systems in which software interacts with the physical world. In the past decades, there has been considerable progress on stability and stabilization of switched systems [Lib03, LA09]. Though, the synthesis of switching controllers for different objectives such as safety or reachability still remains a challenging problem. Controller synthesis for switched systems with safety or reachability specifications can for instance be tackled by direct application of fixed-point computation or dynamic programming using guaranteed over-approximations [ABD$^+$00] or convergent approximations [MT00] of reachable sets. In the first case, the synthesized controllers are correct by design but there is no guarantee that the synthesis algorithm will terminate. In the second case, we can only prove that the synthesized controllers are "correct in the limit" in the sense that correct controllers can be approximated arbitrarily close. The approaches based on the use of symbolic abstractions do not suffer from these drawbacks and can be applied provided we are able to compute a suitable abstraction of the switched system. We show that for a class of incrementally stable switched systems, it is always possible to compute approximately bisimilar symbolic abstractions thus enabling the design of safety and reachability controllers that are correct by design. Moreover, since any precision of approximation can be achieved the maximal or time-optimal controller can be approached arbitrarily close. We also discuss techniques that can help in synthesizing switching controllers of low complexity, using quantization

of the state and an efficient representation of the control map.

In the last part of the chapter, we present an approach for reducing the algorithmic complexity of switching controller synthesis by constructing multi-scale symbolic abstractions that are approximately bisimilar to a switched system. Synthesis algorithms exploiting the specificities of multi-scale abstractions have been implemented in the tool CoSyMA [MGG13]: the abstractions are computed on the fly during controller synthesis and the dynamics at the finest scales are explored only when necessary. We provide experimental results that show significant improvements of the complexity of controller synthesis using multi-scale abstractions.

3.1 Synthesis using Approximately Bisimilar Abstractions

In this section, we show how approximately bisimilar abstractions can be used for the synthesis of safety and reachability controllers. We present our approaches introduced in [Gir12] in the general framework of transition systems. We shall consider only static (i.e. without memory) state-feedback controllers. However, we will just use the term controller for brevity.

Definition 3.1 (Controller) *A controller for transition system $T = (X, U, \mathcal{S}, X^0, Y, \mathcal{O})$ is a set-valued map $\mathcal{C} : X \to 2^U$ such that $\mathcal{C}(x) \subseteq \mathsf{Enab}(x)$, for all $x \in X$. The dynamics of the controlled system is described by the transition system $T/\mathcal{C} = (X, U, \mathcal{S}_\mathcal{C}, X^0, Y, \mathcal{O})$ where the transition map is given by $x' \in \mathcal{S}_\mathcal{C}(x, u)$ if and only if $u \in \mathcal{C}(x)$ and $x' \in \mathcal{S}(x, u)$.*

The support of $\mathcal{C}$ is the set $\mathsf{Supp}(\mathcal{C}) = \{x \in X | \, \mathcal{C}(x) \neq \emptyset\}$. We would like to emphasize the fact that a controller is a set-valued map, at a given state x it enables a set of admissible inputs $\mathcal{C}(x) \subseteq U$. A controller executes as follows: the state x of T is measured, an input $u \in \mathcal{C}(x)$ is selected and actuated; then, the system takes a transition $x' \in \mathcal{S}(x, u)$. The blocking states of $T/\mathcal{C}$ are the elements of $X \setminus \mathsf{Supp}(\mathcal{C})$. For a subset $X' \subseteq X$, we denote $\mathcal{C}(X') = \bigcup_{x \in X'} \mathcal{C}(x)$. Given two approximately bisimilar transition systems, we show in this section how one system can be used for the synthesis of safety or reachability controllers for the other system.

3.1.1 Safety controllers

Problem formulation: Let $T = (X, U, \mathcal{S}, X^0, Y, \mathcal{O})$ be a transition system, let $Y_s \subseteq Y$ be a set of outputs associated with safe states. We consider the synthesis problem that consists in determining a controller that keeps the output of the system inside the specified safe set Y_s.

Definition 3.2 (Safety controller) *A controller $\mathcal{C}$ is a safety controller for T and specification Y_s if for all $x \in \mathsf{Supp}(\mathcal{C})$:*

1. *$\mathcal{O}(x) \in Y_s$ (safety);*
2. *$\forall u \in \mathcal{C}(x)$, $\mathcal{S}(x, u) \subseteq \mathsf{Supp}(\mathcal{C})$ (deadend freedom).*

It is easy to verify from the previous definition that for any initial state $x^0 \in \mathsf{Supp}(\mathcal{C})$, the controlled system $T/\mathcal{C}$ will never reach a blocking state (because of the deadend freedom condition) and its output will remain in the safe set Y_s forever (because of the safety condition).

There are in general several controllers that solve the safety problem. We are usually interested in synthesizing a controller that enables as many actions as possible. This notion of permissivity can be formalized by defining a partial order on controllers.

Definition 3.3 (Maximal safety controller) *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two controllers for transition system T, $\mathcal{C}_1$ is more permissive than $\mathcal{C}_2$, denoted $\mathcal{C}_2 \preceq \mathcal{C}_1$, if for all $x \in X$, $\mathcal{C}_2(x) \subseteq \mathcal{C}_1(x)$. The controller $\mathcal{C}^*$ for T is the maximal safety controller for specification Y_s, if $\mathcal{C}^*$ is a safety controller for specification Y_s, and for all safety controllers $\mathcal{C}$ for specification Y_s, $\mathcal{C} \preceq \mathcal{C}^*$.*

It is well known that the maximal safety controller exists, is unique and can be computed using a fixed point algorithm (see e.g. [Mal02, Tab09]). This algorithm is guaranteed to terminate in a finite number of steps for symbolic transition systems. For other systems, there is no guarantee that the algorithm will terminate. In this case, a synthesis approach based on approximately bisimilar abstractions can help to compute effectively a safety controller with, in addition, an estimation of the distance to maximality.

Abstraction based controller synthesis: Let $T_i = (X_i, U, \mathcal{S}_i, X_i^0, Y, \mathcal{O}_i)$, $i = 1, 2$, be metric transition systems such that $T_1 \sim_\varepsilon T_2$. Let $\mathcal{R}_\varepsilon \subseteq X_1 \times X_2$ denote the approximate bisimulation relation of precision ε between T_1 and T_2. For $x_1 \in X_1$, we denote $\mathcal{R}_\varepsilon(x_1) = \{x_2 \in X_2 | (x_1, x_2) \in \mathcal{R}_\varepsilon\}$. Let T_1 be the system that we want to control and T_2 be an approximately bisimilar abstraction of T_1. We present an approach for synthesizing safety controllers for a specification Y_s.

Definition 3.4 *Let $Y' \subseteq Y$ and $\varphi \geq 0$. The φ-contraction of Y' is the subset of Y defined as follows*

$$C_\varphi(Y') = \left\{ y' \in Y' | \, \forall y \in Y, \; d(y, y') \leq \varphi \implies y \in Y' \right\}.$$

The φ-expansion of Y' is the subset of Y defined as follows

$$E_\varphi(Y') = \left\{ y \in Y | \, \exists y' \in Y', d(y, y') \leq \varphi \right\}.$$

We start by synthesizing a safety controller for the abstraction T_2 and the specification $C_\varepsilon(Y_s)$. This controller is denoted $\mathcal{C}_{2,\underline{\varepsilon}}$. We shall not discuss further the synthesis of this controller which can be done, if T_2 is symbolic, using a fixed point algorithm. The second step of our approach allows us to design a safety controller for system T_1 and specification Y_s, obtained from the controller $\mathcal{C}_{2,\underline{\varepsilon}}$ using the following concretization procedure:

Theorem 3.1 (Correctness by design) *Let $\mathcal{C}_{2,\underline{\varepsilon}}$ be a safety controller for T_2 and specification $C_\varepsilon(Y_s)$. Let $\mathcal{C}_1$ be the controller for T_1 given by*

$$\forall x_1 \in X_1, \; \mathcal{C}_1(x_1) = \mathcal{C}_{2,\underline{\varepsilon}}(\mathcal{R}_\varepsilon(x_1)). \tag{3.1}$$

Then, $\mathcal{C}_1$ is a safety controller for specification Y_s.

If we use the maximal safety controller for T_2, it is desirable to have an estimate of the distance between the controller given by the concretization equation (3.1) and the maximal safety controller for T_1. This is given by the following result:

Theorem 3.2 (Maximality in the limit) *Let $\mathcal{C}^*_{2,\underline{\varepsilon}}$ and $\mathcal{C}^*_{2,\overline{\varepsilon}}$ be the maximal safety controllers for T_2 and specifications $C_\varepsilon(Y_s)$ and $E_\varepsilon(Y_s)$ respectively. Let $\mathcal{C}_1$ and $\mathcal{C}_{1,\overline{2\varepsilon}}$ be the controllers for T_1 obtained by the concretization equation (3.1) from $\mathcal{C}^*_{2,\underline{\varepsilon}}$ and $\mathcal{C}^*_{2,\overline{\varepsilon}}$ respectively. Let $\mathcal{C}^*_1$, $\mathcal{C}^*_{1,\underline{2\varepsilon}}$ and $\mathcal{C}^*_{1,\overline{2\varepsilon}}$ be the maximal safety controllers for T_1 and specifications Y_s, $C_{2\varepsilon}(Y_s)$ and $E_{2\varepsilon}(Y_s)$ respectively. Then,*

$$\mathcal{C}^*_{1,\underline{2\varepsilon}} \preceq \mathcal{C}_1 \preceq \mathcal{C}^*_1 \preceq \mathcal{C}_{1,\overline{2\varepsilon}} \preceq \mathcal{C}^*_{1,\overline{2\varepsilon}}.$$

Hence, by computing the controllers $\mathcal{C}_1$ and $\mathcal{C}_{1,\overline{2\varepsilon}}$ one is able to give a certified upper-bound on the distance between the controller $\mathcal{C}_1$ we will use to control T_1 and the maximal safety controller $\mathcal{C}^*_1$. Moreover, if the safety problem is robust, in the sense that $\mathcal{C}^*_{1,\underline{2\varepsilon}}$ and $\mathcal{C}^*_{1,\overline{2\varepsilon}}$ approach $\mathcal{C}^*_1$ as ε goes to 0 (i.e. slightly different specifications result in only slightly different maximal controllers); then $\mathcal{C}_1$ and $\mathcal{C}_{1,\overline{2\varepsilon}}$ also approach $\mathcal{C}^*_1$ as ε gets smaller and $\mathcal{C}^*_1$ can be approximated arbitrarily close.

3.1.2 Reachability controllers

We present a similar approach for the synthesis of reachability controllers.

Problem formulation: Let $T = (X, U, \mathcal{S}, X^0, Y, \mathcal{O})$ be a transition system, let $Y_s \subseteq Y$ be a set of outputs associated with safe states, let $Y_t \subseteq Y_s$ be a set of outputs associated with target states. We consider the synthesis problem that consists in determining a controller steering the output of the system to Y_t while keeping the output in Y_s along the way. In addition, in order to choose among the possible controllers, we try to minimize the time to reach the target. Thus, we consider an optimal control problem. In this section, we assume for simplicity, that T is non-blocking; it would actually be sufficient to assume that all the states of T associated to outputs in Y_s are non-blocking.

Definition 3.5 (Entry time) *Let $\mathcal{C}$ be a controller for T such that for all $x \in X$, $\mathcal{C}(x) \neq \emptyset$. The entry time of $T/\mathcal{C}$ from $x^0 \in X$ for reachability specification (Y_s, Y_t) is the smallest $N \in \mathbb{N}$ such that for all state trajectories of the controlled system $T/\mathcal{C}$, of length N and starting from x^0, $(x^0, u^0), (x^1, u^1), \ldots, (x^{N-1}, u^{N-1}), x^N$, there exists $K \in \{0, \ldots, N\}$ such that*

1. *$\forall k \in \{0, \ldots, K\}$, $\mathcal{O}(x^k) \in Y_s$;*
2. *$\mathcal{O}(x^K) \in Y_t$.*

The entry time is denoted by $J(T/\mathcal{C}, Y_s, Y_t, x^0)$. If such a $N \in \mathbb{N}$ does not exist, then we define $J(T/\mathcal{C}, Y_s, Y_t, x^0) = +\infty$.

The condition that $\mathcal{C}(x) \neq \emptyset$, for all $x \in X$, ensures that the controlled system $T/\mathcal{C}$ is non-blocking. The states from which the controlled system $T/\mathcal{C}$ is guaranteed to reach Y_t without leaving Y_s are the states with finite entry-time. We can now define the notion of time-optimal controller:

Definition 3.6 (Time-optimal reachability controller) *A controller $\mathcal{C}^*$ for T is a time-optimal reachability controller for specification (Y_s, Y_t) if for all controllers $\mathcal{C}$, for all $x \in X$, $J(T/\mathcal{C}^*, Y_s, Y_t, x) \leq J(T/\mathcal{C}, Y_s, Y_t, x)$. The time-optimal value function for reachability specification (Y_s, Y_t) is defined as $J^*(T, Y_s, Y_t, x) = J(T/\mathcal{C}^*, Y_s, Y_t, x)$.*

Solving the time-optimal control problem consists in synthesizing a time-optimal controller. It is well known that a time-optimal controller exists (but may be not unique) and can be determined using the time-optimal value function (which is unique). The time-optimal value function can be computed using dynamic programming [Ber00, Tab09]. The dynamic programming algorithm is guaranteed to terminate in a finite number of steps for symbolic transition systems. Here again, for other systems, there is no guarantee that the algorithm will terminate and an abstraction-based approach is helpful to compute a sub-optimal controller with an estimation of the distance to optimality.

Abstraction based controller synthesis: Let $T_i = (X_i, U, \mathcal{S}_i, X_i^0, Y, \mathcal{O}_i)$, $i = 1, 2$, be metric transition systems such that $T_1 \sim_\varepsilon T_2$. Let $\mathcal{R}_\varepsilon \subseteq X_1 \times X_2$ denote the approximate bisimulation relation of precision ε between T_1 and T_2. Let T_1 be the system that we want to control and T_2 be an approximately bisimilar abstraction of T_1.

We present an approach for synthesizing reachability controllers. We first synthesize a controller $\mathcal{C}_{2,\underline{\varepsilon}}$ for the abstraction T_2 and the reachability specification given by the contracted safe set $C_\varepsilon(Y_s)$ and target set $C_\varepsilon(Y_t)$. If T_2 is symbolic, this can be done using dynamic programming. Then, we design a controller for T_1 and reachability specification (Y_s, Y_t) using the following concretization procedure:

Theorem 3.3 (Correctness by design) *Let $\mathcal{C}_{2,\underline{\varepsilon}}$ be a controller for T_2, let us define $\mathcal{C}_1$, the controller for T_1 given by*

$$\forall x_1 \in X_1,\ \mathcal{C}_1(x_1) = \mathcal{C}_{2,\underline{\varepsilon}}\left(\arg\min_{x_2 \in \mathcal{R}_\varepsilon(x_1)} J(T_2/\mathcal{C}_{2,\underline{\varepsilon}}, C_\varepsilon(Y_s), C_\varepsilon(Y_t), x_2)\right). \tag{3.2}$$

Then, for all $x_1 \in X_1$:

$$J(T_1/\mathcal{C}_1, Y_s, Y_t, x_1) \leq \min_{x_2 \in \mathcal{R}_\varepsilon(x_1)} J(T_2/\mathcal{C}_{2,\underline{\varepsilon}}, C_\varepsilon(Y_s), C_\varepsilon(Y_t), x_2). \tag{3.3}$$

The previous theorem gives us a way by equation (3.2) to concretize a controller for abstraction T_2 into a controller for T_1. Equation (3.3) provides guarantees on the correctness and the performance of this controller. Particularly, let us remark that the states of $T_1/\mathcal{C}_1$ from which the control objective is achieved (i.e. the states with finite entry-time) are those related through the approximate bisimulation relation $\mathcal{R}_\varepsilon$ to states of $T_2/\mathcal{C}_{2,\underline{\varepsilon}}$ with finite entry-time.

In addition, if $\mathcal{C}_{2,\underline{\varepsilon}}$ is the time-optimal controller for T_2 and reachability specification $(C_\varepsilon(Y_s), C_\varepsilon(Y_t))$, the following result gives estimates of the distance to optimality for the controller $\mathcal{C}_1$.

Theorem 3.4 (Optimality in the limit) *Let $\mathcal{C}^*_{2,\underline{\varepsilon}}$, $\mathcal{C}^*_{2,\overline{\varepsilon}}$ be time-optimal controllers for T_2 and specification $(C_\varepsilon(Y_s), C_\varepsilon(Y_t))$ and $(E_\varepsilon(Y_s), E_\varepsilon(Y_t))$ respectively. Let $\mathcal{C}_1$ be the controller for T_1 obtained from $\mathcal{C}^*_{2,\underline{\varepsilon}}$ by the concretization equation (3.2). Let $\mathcal{C}_{1,\overline{2\varepsilon}}$ be the controller for T_1 given by*

$$\forall x_1 \in X_1,\ \mathcal{C}_{1,\overline{2\varepsilon}}(x_1) = \mathcal{C}^*_{2,\overline{\varepsilon}}\left(\arg\min_{x_2 \in \mathcal{R}_\varepsilon(x_1)} J(T_2/\mathcal{C}^*_{2,\overline{\varepsilon}}, E_\varepsilon(Y_s), E_\varepsilon(Y_t), x_2)\right).$$

Then, for all $x_1 \in X_1$,

$$\begin{aligned} J^*(T_1, E_{2\varepsilon}(Y_s), E_{2\varepsilon}(Y_t), x_1) &\leq J(T_1/\mathcal{C}_{1,\overline{2\varepsilon}}, E_{2\varepsilon}(Y_s), E_{2\varepsilon}(Y_t), x_1) \leq \\ J^*(T_1, Y_s, Y_t, x_1) &\leq J(T_1/\mathcal{C}_1, Y_s, Y_t, x_1) \leq J^*(T_1, C_{2\varepsilon}(Y_s), C_{2\varepsilon}(Y_t), x_1). \end{aligned}$$

By computing the controllers $\mathcal{C}_1$ and $\mathcal{C}_{1,\overline{2\varepsilon}}$ one is able to give a certified upper-bound on the distance to optimality of the controller $\mathcal{C}_1$ we will use to control T_1. Moreover, if the reachability problem is robust (i.e. the time-optimal value function depends continuously on the specification); then $J^*(T_1, Y_s, Y_t, x_1)$ can be approximated arbitrarily close.

In the next section, we will use our approaches for computing safety and reachability controllers for switched systems using approximately bisimilar symbolic abstractions.

3.2 Controllers for Switched Systems

In this section, we apply the previous results to the synthesis of controllers for a class of incrementally stable switched systems for which it is possible to compute approximately bisimilar abstractions.

3.2.1 Approximately bisimilar abstractions of switched systems

We describe an approach, presented in [GPT10], for computing symbolic abstractions of switched systems of the following form:

$$\Sigma : \ \dot{\mathbf{x}}(t) = f_{\mathbf{p}(t)}(\mathbf{x}(t)),\ \mathbf{x}(t) \in \mathbb{R}^n,\ \mathbf{p}(t) \in P$$

where P is a finite set of modes. The switching signals $\mathbf{p} : \mathbb{R}^+ \to P$ are assumed to be piecewise constant functions, continuous from the right and with a finite number of discontinuities on every bounded interval. We use $\mathbf{x}(t, x, \mathbf{p})$ to denote the point reached at time $t \in \mathbb{R}^+$ from the initial condition x under the switching signal $\mathbf{p}$. We assume that the vector fields f_p are locally Lipschitz continuous and such that the trajectories of Σ are defined on $[0, +\infty[$. Necessary and sufficient conditions to be satisfied by f_p can be found in [AS99].

Given a parameter $\tau > 0$, we define a transition system $T_\tau(\Sigma)$ that describes trajectories of duration τ of Σ. This can be seen as a time sampling process. This is natural when the switching in Σ is determined by a periodic controller of period τ. Formally, $T_\tau(\Sigma) = (X, U, \mathcal{S}, X^0, Y, \mathcal{O})$ where the set of states is $X = \mathbb{R}^n$; the set of inputs is the set of modes $U = P$; the transition map is given by

$$\forall x \in \mathbb{R}^n,\ \forall p \in P,\ x' \in \mathcal{S}(x, p) \iff x' = \mathbf{x}(\tau),\ \text{where } \dot{\mathbf{x}}(t) = f_p(\mathbf{x}(t)),\ \mathbf{x}(0) = x;$$

the set of initial states is $X^0 = \mathbb{R}^n$; the set of outputs is $Y = \mathbb{R}^n$; and the observation map $\mathcal{O}$ is the identity map over $\mathbb{R}^n$. $T_\tau(\Sigma)$ is a non-blocking, deterministic, regular metric transition systems when the set of observations $Y = \mathbb{R}^n$ is equipped with the Euclidean distance.

The computation of a symbolic abstraction of $T_\tau(\Sigma)$ can be done by the following approach. We start by approximating the set of states $X = \mathbb{R}^n$ by a lattice:

$$[\mathbb{R}^n]_\eta = \left\{ q \in \mathbb{R}^n \ \middle|\ q_i = k_i \frac{2\eta}{\sqrt{n}},\ k_i \in \mathbb{Z},\ i = 1, \ldots, n \right\},$$

where q_i is the i-th coordinate of q and $\eta > 0$ is a state space discretization parameter. We associate a quantizer $Q_\eta : \mathbb{R}^n \to [\mathbb{R}^n]_\eta$ defined as follows $q = Q_\eta(x)$ if and only if

$$\forall i = 1, \ldots, n,\ q_i - \tfrac{\eta}{\sqrt{n}} \leq x_i < q_i + \tfrac{\eta}{\sqrt{n}}.$$

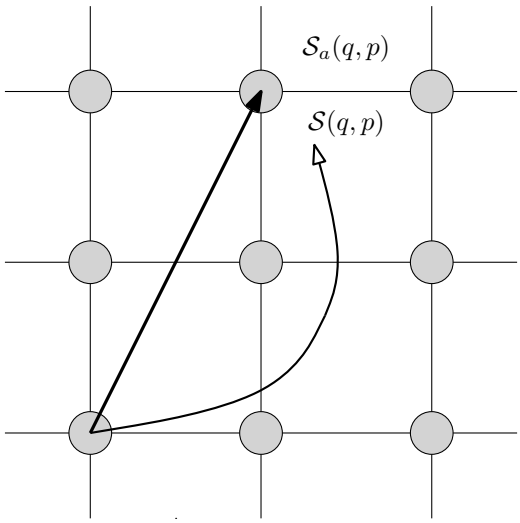

Figure 3.1: Principle for the computation of a symbolic abstraction of a switched system.

By simple geometrical considerations, we can check that for all $x \in \mathbb{R}^n$, $\|Q_\eta(x) - x\| \leq \eta$. We can then define the abstraction of $T_\tau(\Sigma)$ as the transition system $T_{\tau,\eta}(\Sigma) = (X_a, U, \mathcal{S}_a, X_a^0, Y, \mathcal{O}_a)$, where the set of states is $X_a = [\mathbb{R}^n]_\eta$; the set of labels remains the same $U = P$; the transition relation is essentially obtained by quantizing the transition relation of $T_\tau(\Sigma)$ (see Figure 3.1):

$$\forall q \in [\mathbb{R}^n]_\eta,\ \forall p \in P,\ \mathcal{S}_a(q,p) = Q_\eta(\mathcal{S}(q,p));$$

the set of initial states is $X_a^0 = [\mathbb{R}^n]_\eta$; the set of outputs remains the same $Y = \mathbb{R}^n$; and the observation map $\mathcal{O}_a$ is the inclusion map from $[\mathbb{R}^n]_\eta$ to $\mathbb{R}^n$. Note that the transition system $T_{\tau,\eta}(\Sigma)$ is a non-blocking, deterministic, regular metric transition systems when the set of observations $Y = \mathbb{R}^n$ is equipped with the Euclidean norm. It is discrete since its sets of states and actions are respectively countable and finite. Moreover, if we restrict the set of states to a compact subset of $\mathbb{R}^n$, then it can be seen as symbolic.

The approximate bisimilarity of $T_\tau(\Sigma)$ and $T_{\tau,\eta}(\Sigma)$ is related to the notion of incremental stability [Ang02]. Intuitively, a switched system is incrementally stable if the distance between any two trajectories associated with the same switching signal $\mathbf{p}$, but with different initial states, converges asymptotically to 0:

Definition 3.7 (Incremental stability) *The switched system Σ is said to be incrementally globally uniformly asymptotically stable (δ-GUAS) if there exists a $\mathcal{KL}$ function[1] β such that for all $t \in \mathbb{R}^+$, for all $x, y \in \mathbb{R}^n$, for all switching signals $\mathbf{p} \in \mathcal{P}$, the following condition is satisfied:*

$$\|\mathbf{x}(t, x, \mathbf{p}) - \mathbf{x}(t, y, \mathbf{p})\| \leq \beta(\|x - y\|, t). \tag{3.4}$$

[1] A continuous function $\gamma : \mathbb{R}^+ \to \mathbb{R}^+$ is said to belong to class $\mathcal{K}_\infty$ if it is strictly increasing, $\gamma(0) = 0$ and $\gamma(r) \to \infty$ when $r \to \infty$. A continuous function $\beta : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ is said to belong to class $\mathcal{KL}$ if for all fixed s, the map $r \mapsto \beta(r, s)$ belongs to class $\mathcal{K}_\infty$ and for all fixed r, the map $s \mapsto \beta(r, s)$ is strictly decreasing and $\beta(r, s) \to 0$ when $s \to \infty$.

Incremental stability is also similar to the notion of contraction presented in [LS98]. Incremental stability of a switched system can be proved using Lyapunov functions.

Definition 3.8 (Common δ-GUAS Lyapunov function) *A smooth function $\mathcal{V} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ is a common δ-GUAS Lyapunov function for Σ if there exist $\mathcal{K}_\infty$ functions $\underline{\alpha}$, $\overline{\alpha}$ and $\kappa > 0$ such that for all $x_1, x_2 \in \mathbb{R}^n$, for all $p \in P$:*

$$\underline{\alpha}(\|x_1 - x_2\|) \leq \mathcal{V}(x_1, x_2) \leq \overline{\alpha}(\|x_1 - x_2\|);$$
$$\frac{\partial \mathcal{V}}{\partial x_1}(x_1, x_2) \cdot f_p(x_1) + \frac{\partial \mathcal{V}}{\partial x_2}(x_1, x_2) \cdot f_p(x_2) \leq -\kappa \mathcal{V}(x_1, x_2).$$

The existence of a common δ-GUAS Lyapunov function ensures that the switched system Σ is incrementally stable [GPT10]. We need to make the supplementary assumption on the δ-GUAS Lyapunov function that there exists a $\mathcal{K}_\infty$ function γ such that

$$\forall x_1, x_2, x_1', x_2' \in \mathbb{R}^n,\ |\mathcal{V}(x_1, x_2) - \mathcal{V}(x_1', x_2')| \leq \gamma(\|x_1 - x_1'\|) + \gamma(\|x_2 - x_2'\|). \tag{3.5}$$

This assumption is not restrictive provided $\mathcal{V}$ is smooth and we are interested in the dynamics of Σ on a compact subset of $\mathbb{R}^n$, which is often the case in practice. We can now state the main result of the section which establishes the approximate bisimilarity of $T_\tau(\Sigma)$ and $T_{\tau,\eta}(\Sigma)$ under the existence of a common δ-GUAS Lyapunov function.

Theorem 3.5 (Approximately bisimilar abstractions for switched systems) *Let us consider a switched system Σ, time and state sampling parameters $\tau, \eta > 0$ and a desired precision $\varepsilon > 0$. If there exists a common δ-GUAS Lyapunov function $\mathcal{V}$ for Σ such that equation (3.5) holds and*

$$\eta \leq \min\left\{\gamma^{-1}\left((1 - e^{-\kappa\tau})\underline{\alpha}(\varepsilon)\right), \overline{\alpha}^{-1}\left(\underline{\alpha}(\varepsilon)\right)\right\} \tag{3.6}$$

then

$$\mathcal{R}_\varepsilon = \{(x, q) \in \mathbb{R}^n \times [\mathbb{R}^n]_\eta |\ \mathcal{V}(x, q) \leq \underline{\alpha}(\varepsilon)\}$$

is an approximate bisimulation relation of precision ε between $T_\tau(\Sigma)$ and $T_{\tau,\eta}(\Sigma)$. Moreover, $T_\tau(\Sigma) \sim_\varepsilon T_{\tau,\eta}(\Sigma)$.

Let us remark that the δ-GUAS Lyapunov function $\mathcal{V}$ essentially plays the role of bisimulation function here. It should be noted that given a time sampling parameter $\tau > 0$ and a desired precision $\varepsilon > 0$, it is always possible to choose $\eta > 0$ such that equation (3.6) holds. This essentially means that approximately bisimilar discrete abstractions of arbitrary precision can be computed for $T_\tau(\Sigma)$. In order to evaluate the precision of the symbolic abstraction, one needs to compute a common δ-GUAS Lyapunov function; if the dynamics in each mode is affine then a quadratic Lyapunov function may be computed by solving a set of linear matrix inequalities.

The symbolic abstractions can serve for switching controller synthesis using the approaches described in Theorems 3.1 and 3.3. For instance, for a safety property specified by a compact subset $Y_s \subseteq \mathbb{R}^n$, we can use an approximately bisimilar symbolic abstraction $T_{\tau,\eta}(\Sigma)$ of precision ε. We compute $\mathcal{K}_\varepsilon : [\mathbb{R}^n]_\eta \to 2^P$ a safety controller for $T_{\tau,\eta}(\Sigma)$ and specification $C_\varepsilon(Y_s)$. Since there are only a finite number of elements of $[\mathbb{R}^n]_\eta$ in $C_\varepsilon(Y_s)$, the computation is possible in a finite number of steps using a fixed point algorithm. Then, Theorem 3.1 states that the controller given by

$$\forall x \in \mathbb{R}^n,\ \mathcal{C}(x) = \bigcup_{q \in [\mathbb{R}^n]_\eta,\ \mathcal{V}(x,q) \leq \underline{\alpha}(\varepsilon)} \mathcal{K}_\varepsilon(q) \tag{3.7}$$

is a safety controller for $T_\tau(\Sigma)$ and specification Y_s. Theorem 3.3 provides a similar approach for the computation of reachability controllers.

Remark 3.1 *Theorem 3.5 assumes the existence of a common δ-GUAS Lyapunov function. It is actually possible to relax this assumption by considering multiple δ-GUAS Lyapunov functions (one for each mode) and by imposing a minimum dwell time (i.e. the time between two consecutive switches must be greater than a given explicit lower bound). In that case, we have shown in [GPT10] that approximately bisimilar abstraction of arbitrary precision can be computed.*

Application to a boost DC-DC converter: For illustration purpose, we apply our approach to a boost DC-DC converter. It is a switched system with two modes, the two dimensional dynamics associated with both modes are affine of the form $\dot{\mathbf{x}}(t) = A_p\mathbf{x}(t) + b$ for $p = 1, 2$ (see [GPT10] for numerical values). It can be shown that it has a common δ-GUAS Lyapunov function and thus approximately bisimilar abstractions can be computed.

We first consider the problem of regulating the state of the DC-DC converter around a desired nominal state. This can be done for instance by synthesizing a controller that keeps the state of the switched system in a set centered around the nominal state. This is a safety specification. In the following, we consider the specification given by the set $Y_s = [1.1, 1.6] \times [5.4, 5.9]$. We use a time sampling parameter $\tau = 1$ and choose to work with a discrete abstraction that is approximately bisimilar to $T_\tau(\Sigma)$ with precision $\varepsilon = 0.05$. We compute a safety controller for the switched system $T_\tau(\Sigma)$ by the approach described in the previous section. There are a finite number of elements of $[\mathbb{R}^n]_\eta$ in $C_\varepsilon(Y_s)$ (actually 169744) and the fixed point algorithm for the synthesis of the maximal safety controller for the abstraction and specification $C_\varepsilon(Y_s)$ terminates in 2 iterations. The safety controller $\mathcal{C}$ for the switched system $T_\tau(\Sigma)$ and the specification Y_s obtained by the concretization equation (3.1) is shown on the left part of Figure 3.2 where we have represented a trajectory of the system where the switching is controlled using a lazy implementation of the controller $\mathcal{C}$: when the controller has the choice between mode 1 and 2, it keeps the current mode active. We can check that the specification is effectively met. We computed in a similar way the controller $\mathcal{C}_{\overline{2\varepsilon}}$, shown on the right part of Figure 3.2, which gives an upper bound of the maximal safety controller $\mathcal{C}^*$ for switched system $T_\tau(\Sigma)$ and specification Y_s.

We now consider the problem of steering in minimal time the state of the DC-DC converter in the desired region of operation while respecting some safety constraints. This is a time-optimal control problem. We consider the specification given by the safe set $Y_s = [0.65, 1.65] \times [4.95, 5.95]$ and the target set $Y_t = [1.1, 1.6] \times [5.4, 5.9]$. This time, we use a time sampling parameter $\tau = 0.5$ and choose to work with a discrete abstraction that is approximately bisimilar to $T_\tau(\Sigma)$ with precision $\varepsilon = 0.1$. We compute a suboptimal reachability controller for the switched system $T_\tau(\Sigma)$ by the approach described in the previous section. There are only a finite number of elements of $[\mathbb{R}^n]_\eta$ in $C_\varepsilon(Y_s)$ (actually 674041) and the dynamic programming algorithm for the synthesis of the time-optimal controller for the abstraction and reachability specification $(C_\varepsilon(Y_s), C_\varepsilon(Y_t))$ terminates in 94 iterations. The resulting suboptimal controller $\mathcal{C}$ for the switched system $T_\tau(\Sigma)$ for the reachability specification (Y_s, Y_t) is shown on the left part of Figure 3.3 where we have also represented trajectories of the system where the switching is controlled using the synthesized controller. We can check that the specification is effectively met. The entry

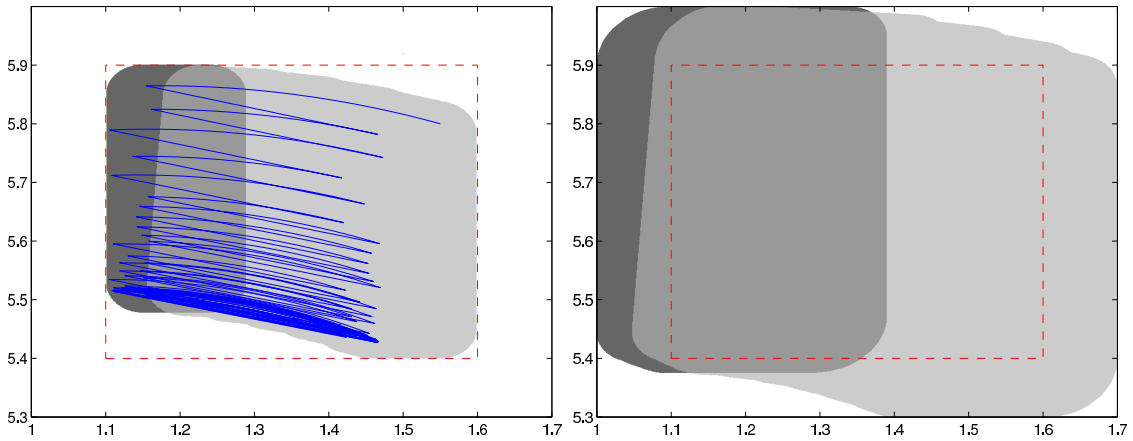

Figure 3.2: Safety controller $\mathcal{C}$ for the system $T_\tau(\Sigma)$ and specification Y_s with controlled trajectory (left); Safety controller $\mathcal{C}_{\overline{2\varepsilon}}$ for the system $T_\tau(\Sigma)$ and specification $E_{2\varepsilon}(Y_s)$ (right); dark gray: mode 1, light gray: mode 2, medium gray: both modes are acceptable, white: no action is allowed. The maximal safety controller $\mathcal{C}^*$ for $T_\tau(\Sigma)$ and specification Y_s satisfies $\mathcal{C} \preceq \mathcal{C}^* \preceq \mathcal{C}_{\overline{2\varepsilon}}$.

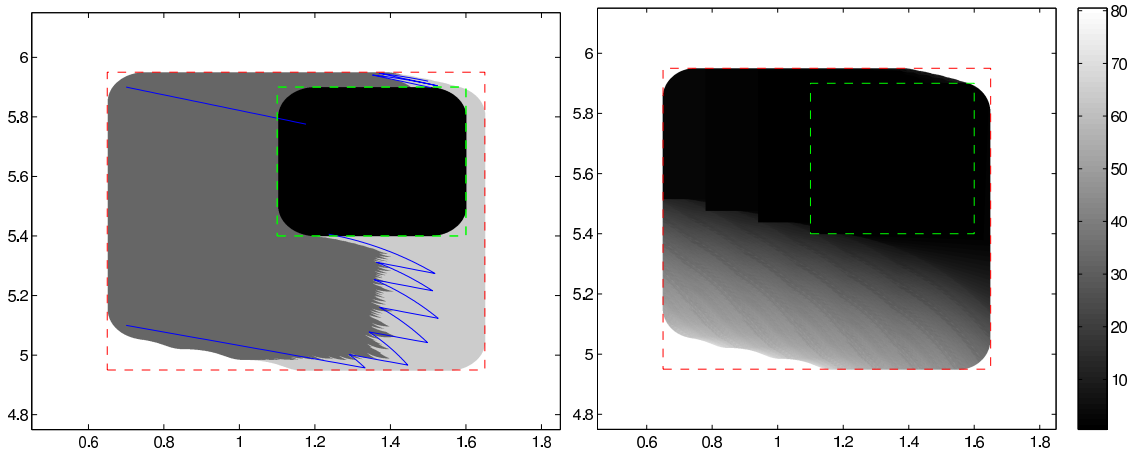

Figure 3.3: Suboptimal controller $\mathcal{C}$ for the switched system $T_\tau(\Sigma)$ and reachability specification (Y_s, Y_t) and trajectories of the controlled switched system (left); Entry-time $J(T_\tau(\Sigma)/\mathcal{C}, Y_s, Y_t, x)$ for the controller $\mathcal{C}$ shown in Figure 3.3 (right).

time associated to $\mathcal{C}$, $J(T_\tau(\Sigma)/\mathcal{C}, Y_s, Y_t, x)$ shown on the right part of Figure 3.3, gives an upper-bound of the time-optimal value function $J^*(T_\tau(\Sigma), Y_s, Y_t, x)$.

3.2.2 Synthesis of low complexity switching controllers

In this paragraph, we go one step further by pursuing the goal of synthesizing low complexity controllers. We focus on safety controllers but a similar work can be done for reachability controllers. Let us consider a switched system Σ and a safety specification given by a compact set $Y_s \subseteq \mathbb{R}^n$. A safety controller $\mathcal{C}$ for $T_\tau(\Sigma)$ is given by (3.7). In

general, for this controller, the union in (3.7) cannot be rigorously pre-computed for all possible values of x but has to be computed online. Moreover, in practice the number of elements to be considered in the union can be quite large (thousands in the previous examples) which result in a significant execution time. This problem would be avoided for controllers that can be written under the form $\mathcal{C} = \mathcal{K} \circ Q_\eta$ where $\mathcal{K} : [\mathbb{R}^n]_\eta \to 2^P$ is a discrete map with finite support and could thus be computed offline. It is possible to synthesize such "quantized" controllers using the following approach presented in [Gir13].

Synthesis of quantized safety controllers: In order to synthesize quantized controllers, we need to establish a new approximation result relating the transition systems $T_\tau(\Sigma)$ and $T_{\tau,\eta}(\Sigma)$.

Proposition 3.1 *Let us consider a switched system Σ, time and state sampling parameters $\tau, \eta > 0$ and a desired precision $\varepsilon > 0$. If there exists a common δ-GUAS Lyapunov function $\mathcal{V}$ for Σ such that equation (3.5) holds and*

$$\varepsilon \geq \eta + \underline{\alpha}^{-1}\left(\frac{2 + e^{-\kappa\tau}}{1 - e^{-\kappa\tau}}\gamma(\eta)\right) \tag{3.8}$$

then

$$\mathcal{R}_\varepsilon = \{(x, q) \in \mathbb{R}^n \times [R^n]_\eta |\ \mathcal{V}(Q_\eta(x), q) \leq \underline{\alpha}(\varepsilon - \eta)\}$$

is an approximate bisimulation relation of precision ε between $T_\tau(\Sigma)$ and $T_{\tau,\eta}(\Sigma)$. Moreover, $T_\tau(\Sigma) \sim_\varepsilon T_{\tau,\eta}(\Sigma)$.

We would like to point out that for given $\tau > 0$ and $\varepsilon > 0$, it is always possible to find $\eta > 0$ such that equation (3.8) holds. Hence, it is possible for any time sampling parameter $\tau > 0$ to compute symbolic models for switched systems of arbitrary precision $\varepsilon > 0$ by choosing a sufficiently small state sampling parameter $\eta > 0$.

We would like to emphasize the differences between Proposition 3.1 and the approximation result presented in Theorem 3.5. The main difference lies in the expression of the approximate bisimulation relation: $(x, q) \in \mathcal{R}_\varepsilon$ if $\mathcal{V}(x, q) \leq \underline{\alpha}(\varepsilon)$ in Theorem 3.5, instead of $\mathcal{V}(Q_\eta(x), q) \leq \underline{\alpha}(\varepsilon - \eta)$ in Proposition 3.1. This difference is fundamental in the sense that it allows us to synthesize quantized controllers. It should also be noted that the relations to be satisfied by the abstraction parameters, τ, η and ε are different: for identical precision and time sampling parameters Proposition 3.1 requires a finer state sampling parameter than Theorem 3.5.

The approach presented in Theorem 3.1 with the approximate bisimulation relation given in Proposition 3.1 leads to the following result:

Proposition 3.2 *Let $\mathcal{K}_\varepsilon : [\mathbb{R}^n]_\eta \to 2^P$ be a safety controller for $T_{\tau,\eta}(\Sigma)$ and specification $C_\varepsilon(Y_s)$. Let $\mathcal{K} : [\mathbb{R}^n]_\eta \to 2^P$ be given by*

$$\forall q \in [\mathbb{R}^n]_\eta,\ \mathcal{K}(q) = \bigcup_{q' \in [\mathbb{R}^n]_\eta,\ \mathcal{V}(q,q') \leq \underline{\alpha}(\varepsilon - \eta)} \mathcal{K}_\varepsilon(q'). \tag{3.9}$$

Then, the map $\mathcal{C} : \mathbb{R}^n \to 2^P$ given by $\mathcal{C} = \mathcal{K} \circ Q_\eta$ is a safety controller for $T_\tau(\Sigma)$ and specification Y_s.

Proposition 3.2 gives an effective way to compute a quantized safety controller for $T_\tau(\Sigma)$. It can be shown that, the support of the discrete map $\mathcal{K}$ is included in $[\mathbb{R}^n]_\eta \cap Y_s$. Since Y_s is compact, the support of the discrete map $\mathcal{K}$ is finite and therefore $\mathcal{K}$ can be pre-computed offline. Then, for a state $x \in \mathbb{R}^n$ the computation of the inputs enabled by $\mathcal{C}$ only requires quantizing the state x and evaluating $\mathcal{K}(Q_\eta(x))$.

Efficient representation of the control law: We now consider the problem of representing the discrete map $\mathcal{K}$ efficiently in order to reduce the memory space needed for the storage of the control law. To reduce the memory needed to store the control law, we will not encode the (set-valued) map $\mathcal{K}$ but a *determinization* of $\mathcal{K}$.

Definition 3.9 (Determinization) *A determinization of the set-valued map $\mathcal{K}$ is a uni-valued map $\mathcal{K}_d : [\mathbb{R}^n]_\eta \cap Y_s \to P$ such that for all $q \in \mathsf{Supp}(\mathcal{K})$, $\mathcal{K}_d(q) \in \mathcal{K}(q)$.*

Let us remark that if $q \notin \mathsf{Supp}(\mathcal{K})$, we do not impose any constraint on the value of $\mathcal{K}_d(q)$. This will gives us more flexibility to reduce the complexity of our control law.

Proposition 3.3 *Let the controller $\mathcal{C}_d : \mathbb{R}^n \to 2^P$ for $T_\tau(\Sigma)$ be given by*

$$\forall x \in \mathbb{R}^n,\ \mathcal{C}_d(x) = \begin{cases} \{\mathcal{K}_d \circ Q_\eta(x)\} & \text{if } Q_\eta(x) \in Y_s \\ \emptyset & \text{otherwise.} \end{cases}$$

Then, for all state trajectories $\sigma_X = (x^0, u^0), (x^1, u^1), \ldots$ of the controlled system $T_\tau(\Sigma)/\mathcal{C}_d$ such that $x^0 \in \mathsf{Supp}(\mathcal{C})$, we have $\mathcal{O}(x^i) \in Y_s$ for all $i = 0, \ldots, \mathsf{l}(\sigma_X)$ and if $\mathsf{l}(\sigma_X) = N \in \mathbb{N}$, x_N is a non-blocking state of $T_\tau(\Sigma)/\mathcal{C}_d$.

The controlled transition system $T_\tau(\Sigma)/\mathcal{C}_d$ is deterministic. It should be noted that the controller $\mathcal{C}_d$ is generally not a safety controller for $T_\tau(\Sigma)$ and specification Y_s in the sense of Definition 3.2 because there might be states in $\mathsf{Supp}(\mathcal{C}_d)$ for which the specification is not met. However, the previous result shows that for an initial state $x^0 \in \mathsf{Supp}(\mathcal{C})$, the controlled system $T_\tau(\Sigma)/\mathcal{C}_d$ will never reach a blocking state and its outputs will remain forever in the safe set Y_s.

We now consider the problem of choosing a determinization $\mathcal{K}_d$ of $\mathcal{K}$ and a representation which requires little memory for its storage. A natural representation for $\mathcal{K}_d$ would be to use a lookup table with a very large number of entries. We propose a more efficient representation inspired by algebraic decision diagrams (ADD's [BFG$^+$93]). The main idea is to use a tree structure which exploits redundant information to represent the map in a more compact way. Also in our case, when $\mathcal{K}(q)$ is empty or when it has more than 2 elements, we have some flexibility for the choice of $\mathcal{K}_d(q)$ which can be used to reduce the size of the representation.

The proposed method for choosing $\mathcal{K}_d$ essentially works as follows: if there exists $p \in P$ such that for all $q \in [\mathbb{R}^n]_\eta \cap Y_s$, $\mathcal{K}(q) = \emptyset$ or $p \in \mathcal{K}(q)$, we can choose $\mathcal{K}_d$ to be the map with constant value p on $[\mathbb{R}^n]_\eta \cap Y_s$. If such an input value does not exists, then we can split the set $[\mathbb{R}^n]_\eta \cap Y_s$ into 2 subsets. The process can then be repeated iteratively: we try to find a suitable constant value on each of the subsets and if this is not possible these sets can be split further. Then, the resulting control law can be naturally represented using a tree structure.

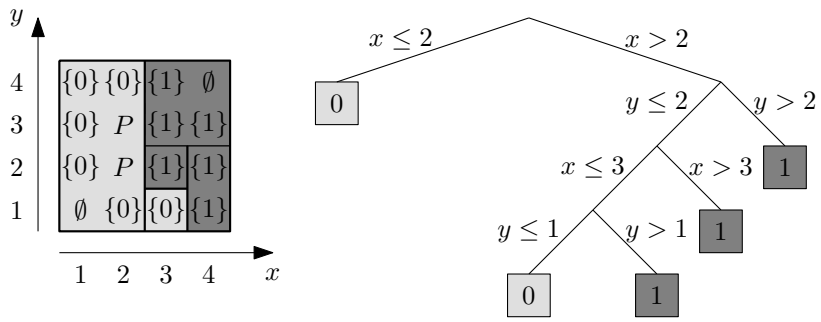

Figure 3.4: A set valued map $\mathcal{K} : \{1,2,3,4\}^2 \to 2^P$ where $P = \{0,1\}$ and a determinization given by colors (dark gray for 1, light gray for 0) and its representation using a tree structure.

Example 3.1 *In Figure 3.4, we show an example of representation using a tree structure of a determinization of a set-valued map* $\mathcal{K} : \{1,2,3,4\}^2 \to 2^P$ *where* $P = \{0,1\}$. *We cannot find a suitable constant value on the whole set* $\{1,2,3,4\}^2$. *Thus, it is split into two subsets* $\{1,2\} \times \{1,2,3,4\}$ *and* $\{3,4\} \times \{1,2,3,4\}$. *For* $q \in \{1,2\} \times \{1,2,3,4\}$ *we can choose* $\mathcal{K}_d(q) = 0$. *On* $\{3,4\} \times \{1,2,3,4\}$, *there is no suitable value. This set is split further into the subsets* $\{3,4\} \times \{1,2\}$ *and* $\{3,4\}^2$. *For* $q \in \{3,4\}^2$, *we can choose* $\mathcal{K}_d(q) = 1$. *On* $\{3,4\} \times \{1,2\}$, *there is no suitable value and this set has to be split futher... By repeating this process, we obtain the determinization* $\mathcal{K}_d$ *represented by the tree structure in Figure 3.4.*

In practice, this process can lead to a very compact representation of the control law as shown in the following example.

Application to a two-room building: For illustration purpose, we consider a simple thermal model of a two-room building:

$$\begin{cases} \dot{T}_1 &= \alpha_{21}(T_2 - T_1) + \alpha_{e1}(T_e - T_1) + \alpha_f(T_f - T_1)p \\ \dot{T}_2 &= \alpha_{12}(T_1 - T_2) + \alpha_{e2}(T_e - T_2) \end{cases}$$

where T_1 and T_2 denote the temperature in each room, $T_e = 10$ is the external temperature and T_f stands for the temperature of a heating device which can be switched on ($p = 1$) or off ($p = 0$). Numerical values of the parameters can be found in [Gir13]. This is a switched system that admits a common δ-GUAS Lyapunov function and therefore our approach can be applied.

We consider the problem of keeping the temperature in the rooms between 20 and 22 degrees Celsius. This is a safety property specified by the safe set $Y_s = [20,22]^2$. We want to use a periodic controller with a period of $\tau = 5$ time units. For the synthesis of the controller, we shall use an approximately bisimilar symbolic abstraction of $T_\tau(\Sigma)$ of precision $\varepsilon = 0.25$. We computed a safety controller $\mathcal{K}_\varepsilon$ for the symbolic abstraction $T_{\tau,\eta}(\Sigma)$ and the specification $\mathrm{Cont}_\varepsilon(Y_s)$. Then, we computed the map $\mathcal{K}$ given by equation (3.9), which is shown in the left part of Figure 3.5. Then, according to Theorem 3.2, the controller $\mathcal{C} = \mathcal{K} \circ Q_\eta$ is a safety controller for $T_\tau(\Sigma)$ and specification Y_s. For a practical implementation of the controller, the storage of the map $\mathcal{K}$ represented by an array would require about 1 million memory units (this is the number of elements in $[\mathbb{R}^2]_\eta \cap Y_s$).

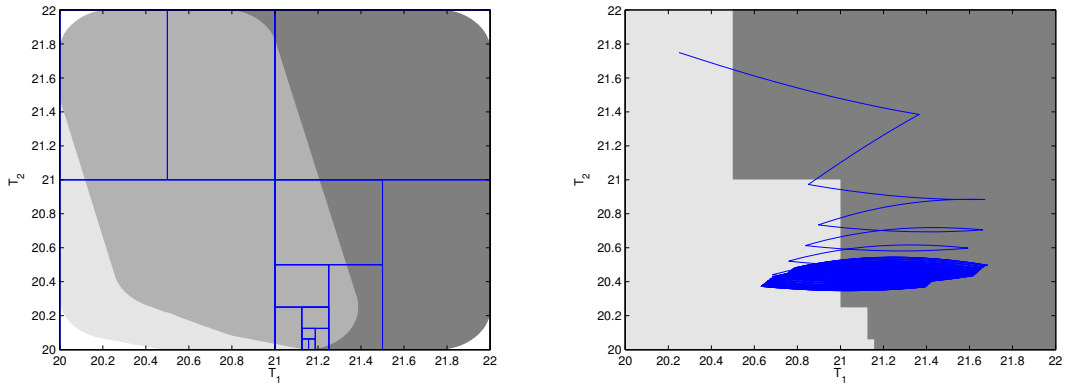

Figure 3.5: Set-valued map $\mathcal{K} : [\mathbb{R}^2]_\eta \cap Y_s \to 2^P$ (white: $\emptyset$, light gray: $\{1\}$, medium gray: P, dark gray: $\{0\}$). The number of elements in $[\mathbb{R}^2]_\eta \cap Y_s$ is about 1 million. In blue, we represented the partition used for the representation of $\mathcal{K}_d$, a determinization of $\mathcal{K}$; the resulting tree structure has only 27 nodes (left). Determinization $\mathcal{K}_d$ of the map $\mathcal{K}$ (light gray: 1, dark gray: 0). In blue, a trajectory of the switched system controlled using the controller $\mathcal{C}_d = \mathcal{K}_d \circ Q_\eta$ (right).

We computed a determinization $\mathcal{K}_d$ of $\mathcal{K}$ following the approach described above. In left part of Figure 3.5, we show the partition used for the representation of $\mathcal{K}_d$, it is to be noted that in each region all values of $\mathcal{K}$ are either $\emptyset$, $\{0\}$, P (which corresponds to value 0 for $\mathcal{K}_d$) or $\emptyset$, $\{1\}$, P (which corresponds to value 1 for $\mathcal{K}_d$). The map $\mathcal{K}_d$ is represented in the right part of Figure 3.5 where we have also represented a trajectory of the switched system controlled using the controller $\mathcal{C}_d = \mathcal{K}_d \circ Q_\eta$. For a practical implementation of the controller, the storage of the map $\mathcal{K}_d$ represented by a tree structure only requires 27 memory units (this is the number of nodes in the tree). We can see with this example that a lot of memory can be saved using an efficient representation and by determinizing the controllers in such a way that their determinization can be represented more compactly. Guarantees of safety for these controllers are still available by Proposition 3.3 which gives insurance of "correctness by design".

3.3 Controller Synthesis using Multi-Scale Abstractions

In the previous section, we have presented two approximation results (Theorem 3.5 and Proposition 3.1) showing that approximately bisimilar abstractions of arbitrary precision could be computed for a class of switched system. In both results, a relation must be satisfied between the desired precision ε, the time sampling parameter τ and the state sampling parameter η. In particular, the smaller τ or ε, the smaller η must be to satisfy equations (3.6) or (3.8). In practice, for a small time sampling parameter τ, the ratio ε/η can be very large and discrete abstractions with an acceptable precision may have a very large number of states as in the examples presented in the previous section.

There are number of applications where the switching has to be fast though this fast switching is generally necessary only on a restricted part of the state space. For instance, for safety controllers, fast switching is needed only when approaching the unsafe set. In

order to enable fast switching while dealing with abstractions with a reasonable number of states, one may consider discrete abstractions enabling transitions of different durations. For transitions of long duration, it is sufficient to consider abstract states on a coarse lattice to meet the desired precision ε. As we consider transitions of shorter durations, it becomes necessary to use finer lattices for the abstract state space. These finer lattices are effectively used only on a restricted area of the state space, where the fast switching is necessary. This allows us to keep the number of states in the abstraction at a reasonable level. This results naturally in a notion of multi-scale abstraction introduced in [CGG11b] and presented in this section.

3.3.1 Multi-scale abstractions for switched systems

To formalize the idea of multi-scale abstraction, we need to change the control paradigm and use self-triggered controllers [VFM03, AT10], where the controller not only determines the mode of the switched system but also the duration during which the mode is to remain active. We assume that the controller can choose from a finite set of durations $\Theta_\tau^N = \{2^{-s}\tau \mid s = 0, \dots, N\}$ that consists of dyadic fractions of a time sampling parameter $\tau > 0$ up to some scale parameter $N \in \mathbb{N}$. The dynamics of the switched system is then naturally described by the transition system $T_\tau^N(\Sigma) = (X, U, \mathcal{S}, X^0, Y, \mathcal{O})$ where the set of states is $X = \mathbb{R}^n$; the set of inputs is $U = P \times \Theta_\tau^N$; the transition relation is given by

$$x' \in \mathcal{S}(x, (p, 2^{-s}\tau)) \iff x' = \mathbf{x}(2^{-s}\tau), \text{ where } \dot{\mathbf{x}}(t) = f_p(\mathbf{x}(t)),\ \mathbf{x}(0) = x;$$

the set of initial states is $X^0 = \mathbb{R}^n$; set of outputs is $Y = \mathbb{R}^n$; the observation map $\mathcal{O}$ is the identity map over $\mathbb{R}^n$. $T_\tau^N(\Sigma)$ is a non-blocking, deterministic, regular metric transition system.

The computation of a symbolic abstraction of $T_\tau^N(\Sigma)$ can then be done using the following approach. We approximate the set of states $X = \mathbb{R}^n$ by the sequence of embedded lattices $[\mathbb{R}^n]_{2^{-s}\eta}$, to which we associate quantizers $Q_{2^{-s}\eta} : \mathbb{R}^n \to [\mathbb{R}^n]_{2^{-s}\eta}$, for $s = 0, \dots, N$. Let us remark that we have $[\mathbb{R}^n]_\eta \subseteq [\mathbb{R}^n]_{2^{-1}\eta} \subseteq \cdots \subseteq [\mathbb{R}^n]_{2^{-N}\eta}$.

Then, we can define the abstraction of $T_\tau^N(\Sigma)$ as the transition system $T_{\tau,\eta}^N(\Sigma) = (X_a, U, \mathcal{S}_a, X_a^0, Y, \mathcal{O}_a)$, where the set of states is $X_a = [\mathbb{R}^n]_{2^{-N}\eta}$; the set of actions remains $U = P \times \Theta_\tau^N$; the transition relation is given by

$$\forall q \in [\mathbb{R}^n]_{2^{-N}\eta},\ \forall (p, 2^{-s}\tau) \in U,\ \mathcal{S}_a(q, (p, 2^{-s}\tau)) = Q_{2^{-s}\eta}(\mathcal{S}(q, (p, 2^{-s}\tau)));$$

the set of initial states $X_a^0 = [\mathbb{R}^n]_\eta$; the set of outputs remains the same $Y = \mathbb{R}^n$; and the observation map $\mathcal{O}_a$ is inclusion map from $[\mathbb{R}^n]_{2^{-N}\eta}$ to $\mathbb{R}^n$. The principle of approximation is illustrated on Figure 3.6. $T_{\tau,\eta}^N(\Sigma)$ is a non-blocking, deterministic, regular metric transition system. It is discrete and can be seen as symbolic if we restrict the set of states to a compact subset of $\mathbb{R}^n$.

It is fundamental to remark that the set of initial states is $[\mathbb{R}^n]_\eta$ and that all the transitions of duration $2^{-s}\tau$ end in states belonging to $[\mathbb{R}^n]_{2^{-s}\eta}$. This means that all trajectories start on the coarsest lattice and that the states on the finer lattices are only accessible by transitions of shorter duration. If $N = 0$, we recover the "uniform" symbolic abstraction $T_{\tau,\eta}(\Sigma)$ presented in Section 3.2.1. It can be shown that under the existence of a common δ-GUAS Lyapunov function and equation (3.5), the transition systems $T_\tau^N(\Sigma)$ and $T_{\tau,\eta}^N(\Sigma)$ are approximately bisimilar:

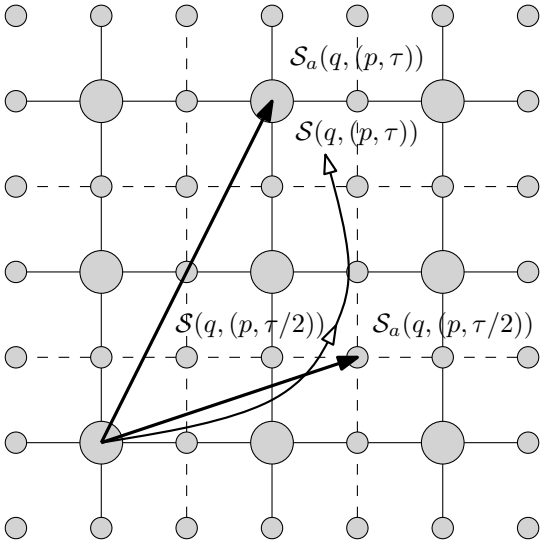

Figure 3.6: Principle for the computation of a multi-scale symbolic abstraction of a switched system. Large and small circles represent elements of $[\mathbb{R}^n]_\eta$ and $[\mathbb{R}^n]_{\eta/2}$, respectively.

Theorem 3.6 (Multi-scale abstractions for switched systems) *Let us consider a switched system Σ, time and state sampling parameters $\tau, \eta > 0$, scale parameter $N \in \mathbb{N}$, and a desired precision $\varepsilon > 0$. Let us assume that there exists a common δ-GUAS Lyapunov function $\mathcal{V}$ for Σ such that equation (3.5) holds and*

$$\eta \leq \min \left\{ \min_{s=0\ldots N} \left[2^s \gamma^{-1} \left((1 - e^{-\kappa 2^{-s}\tau}) \underline{\alpha}(\varepsilon) \right) \right], \overline{\alpha}^{-1} \left(\underline{\alpha}(\varepsilon) \right) \right\} \tag{3.10}$$

then

$$\mathcal{R}_\varepsilon = \left\{ (x,q) \in \mathbb{R}^n \times [\mathbb{R}^n]_{2^{-N}\eta} |\ \mathcal{V}(x,q) \leq \underline{\alpha}(\varepsilon) \right\}$$

is an approximate bisimulation relation of precision ε between $T^N_\tau(\Sigma)$ and $T^N_{\tau,\eta}(\Sigma)$. Moreover, $T^N_\tau(\Sigma) \sim_\varepsilon T^N_{\tau,\eta}(\Sigma)$.

Given a time sampling parameter $\tau > 0$ and a scale parameter $N \in \mathbb{N}$, for all desired precisions $\varepsilon > 0$, there exists $\eta > 0$ such that equation (3.10) holds. This essentially means that approximately bisimilar multi-scale abstractions of arbitrary precision can be computed for $T^N_\tau(\Sigma)$.

3.3.2 Controller synthesis for multi-scale abstractions

We illustrate the use of multi-scale abstractions for synthesizing safety controllers. The main idea is to give the priority to transitions of longer durations so as to keep the state of the abstraction as much as possible at the coarser scales. This motivates us to formalize the notion of lazy safety controllers that always give priority to inputs associated with the longest duration.

Let us consider the multi-scale abstraction $T^N_{\tau,\eta}(\Sigma) = (X_a, U, \mathcal{S}_a, X^0_a, Y, \mathcal{O}_a)$ and a safety specification $Y_s \subseteq Y$. Let $\mathcal{K}^*$ be the maximal safety controller for transition system

$T^N_{\tau,\eta}(\Sigma)$ and specification Y_s. A state $q \in X_a$ is said to be *controllable* with respect to a safety specifications Y_s if $q \in \mathsf{Supp}(\mathcal{K}^*)$. We denote the set of controllable states by $\mathsf{Cont}(Y_s) = \mathsf{Supp}(\mathcal{K}^*)$.

The lazy safety synthesis problem for multi-scale abstractions, introduced in [CGG11a], consists in controlling $T^N_{\tau,\eta}(\Sigma)$ so as to keep any trajectory starting from some initial state in X^0_a within the safe subset of states, while applying in each state a transition of the longest possible duration for which safety can be guaranteed. For that purpose we define priority relations on the set of inputs giving higher priority to transitions of longer duration: for $u, u' \in L = P \times \Theta^N_\tau$ with $u = (p, \theta)$, $u' = (p', \theta')$, $u \prec u'$ if $\theta < \theta'$ and $u \cong u'$ if $\theta = \theta'$.

Definition 3.10 (Maximal lazy safety controller) *A maximal lazy safety controller for $T^N_{\tau,\eta}(\Sigma)$ and specification Y_s is a controller $\mathcal{K}$ such that all controllable states in X^0_a are in $\mathsf{Supp}(\mathcal{K})$, and for all states $x \in \mathsf{Supp}(\mathcal{K})$, x is reachable in $T^N_{\tau,\eta}(\Sigma)/\mathcal{K}$ and the following conditions hold:*

1. *$\mathcal{O}_a(x) \in Y_s$ (safety);*
2. *$\forall u \in \mathcal{K}(x)$, $\mathcal{S}_a(x,u) \subseteq \mathsf{Supp}(\mathcal{K})$ (deadend freedom);*
3. *if $u \in \mathcal{K}(x)$, then for any $u \prec u'$, $\mathcal{S}_a(x,u') \nsubseteq \mathsf{Cont}(Y_s)$ (laziness);*
4. *if $u \in \mathcal{K}(x)$, then for any $u \cong u'$, $u' \in \mathcal{K}(x)$ if and only if $\mathcal{S}_a(x,u) \subseteq \mathsf{Cont}(Y_s)$ (maximality).*

It is clear from conditions 1) and 2) that $\mathcal{K}$ is a safety controller. The controller $\mathcal{K}$ represents a trade-off between maximal permissiveness and efficiency, in the sense that it contains the same initial states as the maximal safety controller; on the other hand, in each state, the enabled transitions are those of maximal duration for which controllability is preserved. Also, $\mathsf{Supp}(\mathcal{K})$ only contains the states that are reachable in $T^N_{\tau,\eta}(\Sigma)/\mathcal{K}$. The fact that $\mathcal{K}$ is safety controller for $T^N_{\tau,\eta}(\Sigma)$ implies that it can be concretized in order to control $T^N_\tau(\Sigma)$ by following the approach presented in Theorem 3.1. The following result shows that the problem of computing a maximal lazy safety controller is well-posed.

Theorem 3.7 (Existence and uniqueness) *There exists a unique maximal lazy safety controller.*

An algorithm for computing the maximal lazy safety controller has been proposed in [CGG11a] and implemented in the tool CoSyMA [MGG13]. It is a fixed point algorithm based on a forward reachability analysis. Starting from the initial states, we explore the trajectories of the system in a depth first search manner using the transitions with longest duration. The transitions of shorter duration are only taken when no other transition leads to a controllable state. In addition, the multi-scale abstraction is computed on the fly so as to keep the number of states in the abstraction as low as possible. This algorithm makes it possible to significantly reduce the algorithmic complexity of controller synthesis as shown in the following.

Application to a boost DC-DC converter: We consider the boost DC-DC converter already presented in the previous section. For the multiscale abstraction $T^N_{\tau,\eta}(\Sigma)$, we consider the safety specification given by $Y_s = [1.15, 1.55] \times [5.45, 5.85]$. We set the

desired precision of abstractions to $\varepsilon = 0.05$. We consider both uniform and multi-scale abstractions to compare the algorithmic complexity of controller synthesis. The uniform abstraction $T_{\tau,\eta}(\Sigma)$ is computed according to the method described in the previous section for time sampling parameters $\tau = 0.5$. The state sampling parameter is chosen according to equation (3.6), that is $\eta = 3 \times 10^{-4}$. We use the multi-scale abstraction $T^N_{\tau,\eta}(\Sigma)$ for $\tau = 32$, $\eta = 0.018$ and $N = 6$ chosen according to Theorem 3.6. This corresponds to transitions of possible duration $\Theta_\tau = \{32, 16, 8, 4, 2, 1, 0.5\}$.

Table 3.1 details the experimental results obtained using the tool CoSyMA for the synthesis of the maximal lazy safety controller for the multi-scale abstraction $T^N_{\tau,\eta}$ as well as the maximal safety controllers for the uniform abstraction $T_{\tau,\eta}$. We report the time needed for the computation of the controller, the size of the abstractions, and the ratio of controllable initial states (that is the ratio $|\mathsf{Supp}(\mathcal{K}) \cap X^0_a|/|X^0_a|$). For the multi-scale controller, we indicate the proportion of inputs associated with each duration. It is worth emphasizing that there is a remarkable reduction in the overall time used to compute the controller using multi-scale abstraction with respect to the use of the uniform one.

	Uniform abstraction $T_{\tau,\eta}(\Sigma)$ $\tau = 0.5$, $\eta = 0.0003$, $\varepsilon = 0.05$
Time	9.2s
Size (10^3)	936
Cont. ratio	93%
	Multi-scale abstraction $T^N_{\tau,\eta}(\Sigma)$ $N = 6, \tau = 32$, $\eta = 0.018$, $\varepsilon = 0.05$
Time	0.6s
Size (10^3)	6
Durations	4 (33%), 2 (9%), 1 (50%), 0.5 (8%)
Cont. Ratio	92%

Table 3.1: Experimental results for the Boost DC-DC Converter

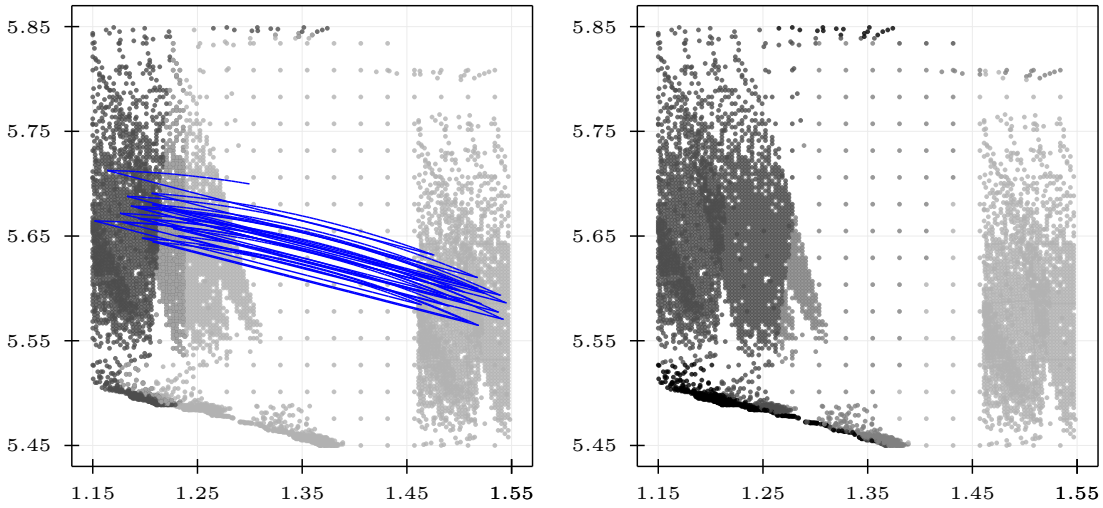

Figure 3.7: The maximal lazy safety controller for $T^N_{\tau,\eta}(\Sigma)$. Mode map (left) - dark gray : mode 1, light gray : mode 2, medium grey : both modes are enabled; Duration map (right) - light gray : 4, medium gray : 2, dark gray : 1, black : 0.5.

Interestingly, this reduction in computation time and abstraction size does not affect the performance of the multi-scale controller, which yields a ratio of controllable initial states similar to the uniform counterpart. Figure 3.7 depicts the controller computed by our algorithm for $T^N_{\tau,\eta}(\Sigma)$.

Discussion: The results presented in this chapter were developed within the ANR projects VAL-AMS and VEDECY and the UJF MSTIC project SYMBAD.

The results presented in Section 3.1 were published in [Gir12]. The main results on the construction of approximately bisimilar symbolic abstractions, presented in Section 3.2.1 were obtained in collaboration with Giordano Pola, University of L'Aquila, and Paulo Tabuada, University of California at Los Angeles, and published in [GPT10]. The techniques for the synthesis of low complexity switching controllers have been presented in [Gir13]. The work on multiscale abstractions has been done in collaboration with Javier Càmara and Sebti Mouelhi during their postdoctoral stay at INRIA Rhône-Alpes under the co-supervision of Gregor Goessler and myself. The results presented in Section 3.3 have been published in [CGG11b, CGG11a, MGG13].

In Section 3.1, we have presented concretization procedures that are specific to safety and reachability controllers. Actually, there exists a "natural" concretization procedure described in [Tab09, Gir10] that can be applied to any type of controllers and that essentially renders the two controlled systems approximately bisimilar. However, the controller for the concrete system obtained via this concretization procedure is a dynamic state-feedback controller (i.e. the controller has a memory) when it is known that for safety or reachability, it is sufficient and optimal to consider static state-feedback controllers. Controller synthesis using approximately bisimilar abstractions has also been considered in [TI08, MT10]. In [TI08], the authors use approximately bisimilar abstractions to design a suboptimal controller for a fixed bounded horizon optimal control problem. Our work is more closely related to [MT10] where time-optimal control is considered as well but where controllers are concretized through the natural procedure.

We would also like to mention the connections between the problems considered in this chapter and some problems in viability theory [Aub01]. The notions of safety and maximal safety controllers are clearly related to that of viability domains and kernels, respectively. Hence, the approach described in this chapter offers an alternative to the viability kernel algorithm [SP94] for computing the viability kernel of an incrementally stable system. Similarly, the reachability problem studied in this chapter can be naturally confronted to the notion of viable capture basin.

There has been a significant work on computing symbolic abstractions for various class of systems using approximate simulation or bisimulation. The earliest results deal with linear systems and were proposed independently by Pola and Tabuada in [Tab06, PT07] and myself [Gir07]. These results were then extended to switched systems [GPT10], nonlinear systems with or without disturbances [PGT08, PT09], time delay systems [PPBT10], networked control systems [BPB12] and stochastic systems [ZMM$^+$13]. The concept of approximately bisimilar multi-scale abstractions has also been explored in [TI09] where the multi-scale feature is used for accommodating locally the precision of the abstraction while the time sampling period remains constant. All these results require an assumption related to incremental stability of the considered systems. Actually, we have shown in [PGT08], for non-linear systems, that incremental stability is a necessary and sufficient condition for approximate bisimilarity of the original system and abstractions defined us-

ing a quantization of the continuous dynamics on a uniform lattice. A recent result makes it possible to relax this assumption by considering the notion of alternating approximate simulation relations [ZPJT12]. Let us remark that our techniques for controller synthesis can be used with all the symbolic abstractions, mentioned above, possibly with some adaptation of our results.

Chapter 4

Other Applications of Approximate Simulation

Résumé : *Le cadre d'approximation présenté au Chapitre 2 a des applications au delà des techniques de synthèse de contrôleurs présentées dans le chapitre précédent. Dans ce chapitre, nous décrivons brièvement, trois autres applications des notions de simulation et bisimulation approchées. Dans la première partie, nous montrons comment la notion de fonction de simulation peut être utilisée pour relier formellement les comportements de deux systèmes dynamiques continus afin de concevoir des systèmes hiérarchiques de commande [GP09]. Dans la deuxième partie, nous présentons une caractérisation des relations de simulation approchée utile pour définir des approximations de systèmes hybrides [GJP08]. Dans la dernière partie, nous montrons comment les fonctions d'auto-bisimulation (c'est à dire une fonction de bisimulation entre un système et lui même) permettent de concevoir des algorithmes de vérification qui ne nécessitent que le calcul d'un nombre fini de trajectoires [FGP06]. Dans les trois cas, la notion de fonction de simulation ou de bisimulation est légèrement adaptée au problème étudié, cependant à chaque fois l'esprit du cadre d'approximation présenté dans le Chapitre 2 est préservé.*

The approximation framework presented in Chapter 2 has applications besides the controller synthesis techniques using approximately bisimilar symbolic abstractions, presented in the previous chapter. In this chapter, we briefly present three other applications of approximate simulation or bisimulation. In the first part, we show how the notion of simulation function can be used to relate two continuous control systems in order to design hierarchical controllers. In the second part, we present an effective characterization of approximate simulation relations that is useful for defining approximations of hybrid systems. In the last part, we show how auto-bisimulation functions (that are bisimulation functions between a system and itself) allow us to propose verification algorithms that require computing only a finite number of trajectories of the system.

In all cases, the notion of simulation or bisimulation function has been adapted to the considered setting and slightly differs from the Definitions 2.6 and 2.7. However, the philosophy remains the same: a simulation or bisimulation function is a function bounding the distance between outputs and decreasing during the evolution of the systems.

4.1 Hierarchical Control Design using Simulation Functions

Controlling complex systems in order to achieve sophisticated tasks constitutes a great challenge of system engineering. Handling at once both complexities of the dynamics and of the specification may lead to untractable problems and therefore a hierarchical approach to controller synthesis is often highly desirable. A hierarchical control architecture has (at least) two layers. The first layer consists of a precise (and complex) model of the plant that need to be controlled and is usually referred to as the *concrete system*. The second layer consists of a coarse (and simple) model of the plant that is used for control synthesis and is referred to as the *abstract system* or *abstraction*. The main challenge of such approaches is the refinement of control laws designed for the abstract system in order to control the concrete system.

In this section, we present our results from [GP09] introducing a hierarchical control framework based on the notion of approximate simulation. Given a complex system that need to be controlled and a simpler abstraction, we show how the knowledge of a simulation function allows us to synthesize hierarchical control laws. For the class of linear control systems, we give an effective characterization of the simulation functions allowing us to use algorithmic procedures for their computation.

4.1.1 Hierarchical control architecture

Let us consider the control systems given, for $j \in \{1, 2\}$ by:

$$\Sigma_j : \begin{cases} \dot{\mathbf{x}}_j(t) &= f_j(\mathbf{x}_j(t), \mathbf{u}_j(t)), & \mathbf{x}_j(t) \in \mathbb{R}^{n_j},\ \mathbf{u}_j(t) \in \mathbb{R}^{p_j} \\ \mathbf{y}_j(t) &= g_j(\mathbf{x}_j(t)), & \mathbf{y}_j(t) \in \mathbb{R}^{k}. \end{cases}$$

where $\mathbf{u}_j : \mathbb{R}^+ \to \mathbb{R}^{p_j}$, $\mathbf{x}_j : \mathbb{R}^+ \to \mathbb{R}^{n_j}$ and $\mathbf{y}_j : \mathbb{R}^+ \to \mathbb{R}^k$ are input, state and output trajectories of Σ_j. We assume that the vector field f_j is such that for any measurable input trajectory $\mathbf{u}_j$, for any initial condition in $\mathbb{R}^{n_j}$, there exist unique state and output trajectories (see e.g. [AS99] for necessary and sufficient conditions).

We refer to Σ_1 as the *concrete system*, that is the (complex) system that we actually want to control. Control is synthesized hierarchically, using the *abstract system* Σ_2, giving a simpler, though less precise, description of the dynamics of the system. Note that systems Σ_1 and Σ_2 have the same output space, but may have different state and input spaces. In particular, the fact that we have different input spaces differs from the work presented in the two previous chapters. Since we have different input spaces, we cannot ask for equality of inputs as previously.

The proposed hierarchical control approach allows us to refine inputs designed for the abstract system Σ_2 in order to control the concrete system Σ_1. We adapt the definition of simulation function given previously to the specific case of continuous control systems. Essentially, a simulation function of Σ_2 by Σ_1 is a function bounding the distance between the system outputs and such that for any input of Σ_2, there exists an input of Σ_1 (given by an *interface*) that makes the function decrease.

Definition 4.1 (Simulation function and interface) *Let $\mathcal{V} : \mathbb{R}^{n_2} \times \mathbb{R}^{n_1} \to \mathbb{R}^+$ be a smooth function and $u_{\mathcal{V}} : \mathbb{R}^{p_2} \times \mathbb{R}^{n_2} \times \mathbb{R}^{n_1} \to \mathbb{R}^{p_1}$ be a continuous function. $\mathcal{V}$ is a simulation function of Σ_2 by Σ_1 and $u_{\mathcal{V}}$ is an associated interface if there exists a $\mathcal{K}$*

function[1] γ *such that for all* $(x_2, x_1) \in \mathbb{R}^{n_2} \times \mathbb{R}^{n_1}$,

$$\mathcal{V}(x_2, x_1) \geq \|g_1(x_1) - g_2(x_2)\| \tag{4.1}$$

and for all $u_2 \in \mathbb{R}^{p_2}$, *satisfying* $\gamma(\|u_2\|) < \mathcal{V}(x_2, x_1)$,

$$\frac{\partial \mathcal{V}(x_2, x_1)}{\partial x_2} \cdot f_2(x_2, u_2) + \frac{\partial \mathcal{V}(x_2, x_1)}{\partial x_1} \cdot f_1(x_1, u_{\mathcal{V}}(u_2, x_2, x_1)) < 0 \tag{4.2}$$

It is interesting to note that the notion of interface can be dropped if one adopts the formalism of alternating approximate simulation [PT09]. As previously stated, a simulation function allows us to bound the distance between outputs of Σ_2 and Σ_1 when the input of Σ_1 is obtained from that of Σ_2 through the interface:

Proposition 4.1 *Let* $\mathcal{V}$ *be a simulation function of* Σ_2 *by* Σ_1 *and* $u_{\mathcal{V}}$ *an associated interface. Let* $\mathbf{u}_2 : \mathcal{I} :\to \mathbb{R}^{p_2}$ *with* $0 \in \mathcal{I} \subseteq \mathbb{R}^+$ *be an input trajectory of* Σ_2, *let* $\mathbf{x_2}$ *and* $\mathbf{y_2}$ *be associated state and output trajectories of* Σ_2. *Let* $\mathbf{x_1}$ *be a state trajectory of* Σ_1 *satisfying the differential equation*

$$\dot{\mathbf{x}}_1(t) = f_1\left(\mathbf{x}_1(t), u_{\mathcal{V}}(\mathbf{u}_2(t), \mathbf{x}_2(t), \mathbf{x}_1(t))\right)$$

and let $\mathbf{y_1}$ *be the associated output trajectory. Then, for all* $t \in \mathcal{I}$,

$$\|\mathbf{y}_2(t) - \mathbf{y}_1(t)\| \leq \max\left\{\mathcal{V}(\mathbf{x}_2(0), \mathbf{x}_1(0)), \gamma(\|\mathbf{u}_2\|_\infty)\right\}.$$

The control architecture, allowing us to refine the inputs of the abstract system through the interface $u_{\mathcal{V}}$ is shown on Figure 4.1. The applicability of our approach relies on our capability of computing a simulation function and an associated interface. In the following section, we give an effective characterization of simulation functions for linear control systems allowing us to design algorithmic procedures for their computation.

4.1.2 Simulation functions for linear systems

In the following, we assume that the concrete and abstract systems are linear control systems, for $j \in \{1, 2\}$:

$$\Sigma_j : \left\{ \begin{array}{rcll} \dot{\mathbf{x}}_j(t) & = & A_j \mathbf{x}_j(t) + B_j \mathbf{u}_j(t) & \mathbf{x}_j(t) \in \mathbb{R}^{n_j},\ \mathbf{u}_j(t) \in \mathbb{R}^{p_j} \\ \mathbf{y}_j(t) & = & C_j \mathbf{x}_j(t) & \mathbf{y}_j(t) \in \mathbb{R}^k \end{array} \right. \tag{4.3}$$

We assume, without loss of generality, that $\mathsf{Rank}(B_1) = p_1$ and $\mathsf{Rank}(C_1) = k$. We further assume that the concrete system Σ_1 is stabilizable. Then, there exists a $p_1 \times n_1$ matrix K such that the matrix $A_1 + B_1 K$ is Hurwitz. Then, there also exist a positive definite symmetric matrix M and a strictly positive scalar number λ such that the following matrix inequalities hold:

$$M \geq C_1^T C_1,$$
$$(A_1 + B_1 K)^T M + M(A_1 + B_1 K) \leq -2\lambda M.$$

Let us remark that the stabilizing gain K and the matrix M can be computed jointly using semidefinite programming. We now give an effective characterization of simulation functions and of the associated interfaces.

[1] A function $\gamma : \mathbb{R}^+ \to \mathbb{R}^+$ is a $\mathcal{K}$ function if it is continuous, strictly increasing and satisfies $\gamma(0) = 0$.

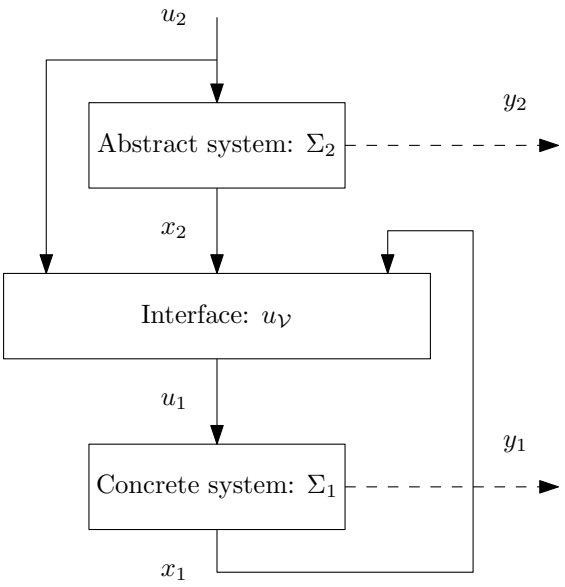

Figure 4.1: Hierarchical control system architecture.

Theorem 4.1 (Simulation functions for linear systems) *Let us assume that there exists an $n_1 \times n_2$ matrix P and a $p_1 \times n_2$ matrix Q such that the following linear matrix equations hold:*

$$PA_2 = A_1P + B_1Q, \tag{4.4}$$

$$C_2 = C_1P. \tag{4.5}$$

Then, the function defined by

$$\mathcal{V}(z,x) = \sqrt{(Px_2 - x_1)^T M (Px_2 - x_1)}$$

is a simulation function of Σ_2 by Σ_1 and an associated interface is given by

$$u_{\mathcal{V}}(u_2, x_2, x_1) = Ru_2 + Qx_2 + K(x_1 - Px_2).$$

where R is an arbitrary $p_1 \times p_2$ matrix. The $\mathcal{K}$ function γ such that equation (4.2) holds is given by

$$\gamma(\nu) = \left\| \sqrt{M}(B_1R - PB_2) \right\| \nu/\lambda.$$

It is minimal for $R = (B_1^T M B_1)^{-1} B_1^T M P B_2$.

Choosing the matrix R in order to minimize the function γ is important as it allows us to tighten the bound on the distance between output trajectories given by Proposition 4.1. Let us remark that equations (4.4) and (4.5) imposes conditions on the matrices A_2 and C_2 of the abstraction. Moreover, if we can choose the matrix Q to be zero then we can guarantee that a uniformly bounded input $\mathbf{u}_2$ of Σ_2 results in a uniformly bounded input $\mathbf{u}_1$ for Σ_1. In general the abstraction is not given a priori and is part of the design parameters; in [GP09], we proposed an approach for the computation of a suitable abstraction such that these matrix equations are satisfied and our approach can be applied. In the following example, we show an application of our hierarchical control framework.

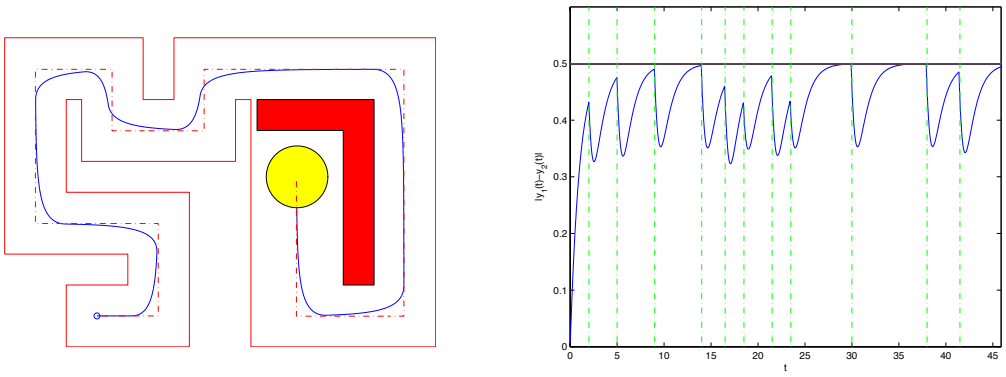

Figure 4.2: Output trajectory $\mathbf{y}_1$ (plain) of the concrete system Σ_1 and output trajectory $\mathbf{y}_2$ (dashed) of the abstract system Σ_2 (left). Value of $\|\mathbf{y}_1(t) - \mathbf{y}_2(t)\|$ for the trajectories of Σ_1 and Σ_2 presented on the left part of the figure. We can see that it is bounded by $= 0.5$ (horizontal line). The vertical lines correspond to the times at which the direction of the trajectory of the abstract system Σ_2 changes (right).

Example 4.1 *We consider a second order model of a mobile robot in a plane:*

$$\Sigma_1 : \ \ddot{\mathbf{y}}_1(t) = \mathbf{u_1}(t),\ \mathbf{y}_1(t) \in \mathbb{R}^2,\ \mathbf{u}_1(t) \in \mathbb{R}^2$$

where $\mathbf{y}_1(t)$ *is the position of the robot. We want to use for control design a first order abstraction:*

$$\Sigma_2 : \ \dot{\mathbf{y}}_2(t) = \mathbf{u_2}(t),\ \mathbf{y}_2(t) \in \mathbb{R}^2,\ \mathbf{u}_2(t) \in \mathbb{R}^2.$$

These systems can be written under the form (4.3) with $\mathbf{x}_1(t) = (\mathbf{y}_1(t), \dot{\mathbf{y}}_1(t))$ *and* $\mathbf{x}_2(t) = \mathbf{y}_2(t)$*. Applying the approach presented above, we designed a simulation function of* Σ_2 *by* Σ_1 *given by*

$$\mathcal{V}(y_2, y_1, \dot{y}_1) = \sqrt{\|y_2 - y_1\|^2 + 8890.1\|y_2 - y_1 - \dot{y}_1\|^2}.$$

The proposed interface is

$$u_{\mathcal{V}}(u_2, y_2, y_1, \dot{y}_1) = u_2 - 1.0006(y_1 - y_2) - 2.0005\dot{y}_1.$$

The associated function γ *is* $\gamma(\nu) = \nu$*. We consider the problem of driving the robot in the environment shown in the left part of Figure 4.2. It consists of a corridor of width* 1*. At the end of the corridor, there is a room with an obstacle. The goal of the motion planning problem consists in reaching a target which is a circle of diameter* 1*, behind the obstacle. Since the abstract system* Σ_2 *is fully actuated, it is easy to synthesize a path for this system. This path is represented by the dashed line in left part of Figure 4.2. It is clear that any trajectory remaining within distance* 0.5 *from this path satisfies the problem specification. We assume that initially* $\mathbf{y}_1(0) = \mathbf{y}_2(0)$ *and* $\dot{\mathbf{y}}_1(0) = 0$*. We use inputs of norm* 0.5 *for the abstraction* Σ_2*, then, from Proposition 4.1, we know that the ouput trajectory* $\mathbf{y}_1$*, obtained by connecting the abstract system and the concrete system through the interface, remains within distance* 0.5 *of* $\mathbf{y}_2$*, and thus satisfies the specification of the*

motion planning problem. The output trajectory $\mathbf{y}_1$ *is represented by the full line in the left part of Figure 4.2. It is clear that it satisfies the specification of the motion planning problem. In the right part of Figure 4.2, we represented the evolution of* $\|\mathbf{y}_1(t)-\mathbf{y}_2(t)\|$*, we can check that it remains bounded by* 0.5 *which is expected from Proposition 4.1. Moreover, we can see that this bound is tight.*

4.2 Approximation of Hybrid Systems

Approximation techniques are crucial for the analysis or synthesis of complex hybrid systems. Most of the proposed approaches are based on the guaranteed over-approximation of the dynamics of a given hybrid system using the notion of exact simulation and bisimulation relations [HHWT98, PvdSB04, HTP05, ADG07, GM12]. However, for hybrid systems, typically observed over the real numbers, approximation notions based on the distance between behaviors is more natural than those based on behavior inclusion. In this section, we present our work from [GJP08], applying the notion of approximate simulation relations for approximation of hybrid systems. Using simulation functions, we develop a characterization of approximate simulation relations which can be used for hybrid systems approximation.

4.2.1 Hybrid systems as transition systems

In this section, we introduce the rather general class of hybrid systems with outputs that we consider and define the transition systems describing their dynamics.

Definition 4.2 (Hybrid automaton) *A hybrid automaton is a tuple* $\Sigma = (L, n, p, E, C, I, G, R, X^0)$ *where*

- L *is a finite set of locations or modes.*
- $n \in \mathbb{N}$ *is the dimension of the continuous state space. The set of states of the hybrid system is* $X = L \times \mathbb{R}^n$*.*
- $p \in \mathbb{N}$ *is the dimension of the continuous output space. The set of outputs of the hybrid system is* $Y = L \times \mathbb{R}^p$*.*
- $E \subseteq L \times L$ *is the set of events or discrete transitions.*
- $C = \{C_l |\ l \in L\}$ *defines the continuous dynamics for each location. For each* $l \in L$*,* C_l *is a triple* (f_l, g_l, D_l) *where* $f_l : \mathbb{R}^n \times D_l \to \mathbb{R}^n$*,* $g_l : \mathbb{R}^n \to \mathbb{R}^p$ *are continuous maps and* $D_l \subseteq \mathbb{R}^{m_l}$ *is a compact set of inputs which should be seen as disturbances accounting for modeling uncertainties rather than control inputs. We assume that* f_l *is Lipschitz continuous and that for each* $x \in \mathbb{R}^n$*,* $f_l(x, D_l)$ *is a compact convex set. While the discrete part of the state is* l*, the continuous variables (i.e. the continuous part* x *of the state and the continuous part* y *of the output) evolve according to*

$$\begin{cases} \dot{\mathbf{x}}(t) &= f_l(\mathbf{x}(t), \mathbf{d}(t)),\ \mathbf{d}(t) \in D_l \\ \mathbf{y}(t) &= g_l(\mathbf{x}(t)). \end{cases}$$

- $I = \{I_l |\ l \in L\}$ *defines an invariant set for each location. For each* $l \in L$*,* $I_l \subseteq \mathbb{R}^n$ *constrains the value of the continuous part of the state while the discrete part is* l*.*

- $G = \{G_e |\ e \in E\}$ *defines the guard for each discrete transition. For each* $e = (l, l') \in E$, $G_e \subseteq I_l$. *The discrete transition* e *is enabled when the continuous part of the state is in* G_e.
- $R = \{R_e |\ e \in E\}$ *defines the reset map for each discrete transition. For each* $e = (l, l') \in E$, $R_e : G_e \to 2^{I_{l'}}$ *has compact images. When the event* e *occurs, the continuous part of the state is reset using the map* R_e.
- $X^0 \subseteq X$ *is the set of initial states:* $X^0 = \bigcup_{l \in L} \{l\} \times I_l^0$, *where* $I_l^0 \subseteq I_l$ *are compact sets.*

The semantics of a hybrid system is well established and will become clear with the definition of the transition system associated to Σ. In the spirit of [ACH+95], we describe the dynamics of Σ by the nondeterministic transition system $T(\Sigma) = (X, U, \mathcal{S}, X^0, Y, \mathcal{O})$ where the set of states X, the set of outputs Y, and the set initial states X^0 are the same as in the hybrid system Σ. The set of inputs is $U = \mathbb{R}^+ \cup \{\tau\}$ where the labels in $\mathbb{R}^+$ represent the durations labeling the continuous transitions while the symbol τ is used to label discrete transitions occurring instantaneously. The ouput map is defined naturally by $\mathcal{O}(l, x) = (l, g_l(x))$. The transition map $\mathcal{S}$ is given by:

1. *continuous transitions* : For $t \in \mathbb{R}^+$, $(l, x') \in \mathcal{S}((l, x), t)$ if and only if there exists a measurable function $\mathbf{d}$ and an absolutely continuous function $\mathbf{x}$ such that $\mathbf{x}(0) = x$, $\mathbf{x}(t) = x'$ and for almost all $s \in [0, t]$,
$$\dot{\mathbf{x}}(s) = f_l(\mathbf{x}(s), \mathbf{d}(s)), \text{ with } \mathbf{d}(s) \in D_l \text{ and } \mathbf{x}(s) \in Inv_l \ .$$
2. *discrete transitions* : $(l', x') \in \mathcal{S}((l, x), \tau)$ if and only if $(l, l') = e \in E$, $x \in G_e$ and $x' \in R_e(x)$.

The transition system is metric when the set outputs Y of the hybrid system Σ is equipped with the following metric d:
$$d\left((l_1, y_1), (l_2, y_2)\right) = \begin{cases} \|y_1 - y_2\|, & \text{if } l_1 = l_2 \\ +\infty, & \text{if } l_1 \neq l_2 \end{cases}$$
where $\|.\|$ is the usual Euclidean norm. In the following, we give a characterization of approximate simulation relations, suitable for hybrid systems; showing that the approximation framework presented in Chapter 2 can be applied in an effective way to hybrid systems.

4.2.2 Approximate simulation relations for hybrid systems

Let $\Sigma_i = (L_i, n_i, p_i, E_i, C_i, I_i, G_i, R_i, X_i^0)$, $(i = 1, 2)$ be two hybrid systems with the same sets of outputs (i.e. $L_1 = L_2 = L$ and $p_1 = p_2 = p$). Let $T(\Sigma_i) = (X_i, U, \mathcal{S}_i, Y, X_i^0)$, $(i = 1, 2)$ be the associated transition systems, they have the same sets of inputs $U = \mathbb{R}^+ \cup \{\tau\}$ and outputs $Y = L \times \mathbb{R}^p$ and therefore we can apply the approximation techniques presented in Chapter 2. In the following, we focus on the approximation of the continuous dynamics of hybrid systems, so we will assume that the discrete dynamics of both systems are the same (i.e. $E_1 = E_2 = E$). In this section, we provide a characterization of approximate simulation relations thus establishing sufficient conditions such that $T(\Sigma_2)$ approximately simulates $T(\Sigma_1)$. It uses the notion of simulation function that we define, in this context, as follows:

Definition 4.3 (Simulation function) *A smooth function $\mathcal{V}_l : \mathbb{R}^{n_{1,l}} \times \mathbb{R}^{n_{2,l}} \to \mathbb{R}^+$ is a simulation function of $C_{1,l}$ by $C_{2,l}$ if for all $(x_1, x_2) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$, the following equations hold*

$$\mathcal{V}_l(x_1, x_2) \geq \|g_{1,l}(x_1) - g_{2,l}(x_2)\|, \tag{4.6}$$

$$\sup_{d_1 \in D_{1,l}} \inf_{d_2 \in D_{2,l}} \left(\frac{\partial \mathcal{V}_l(x_1, x_2)}{\partial x_1} f_{1,l}(x_1, d_1) + \frac{\partial \mathcal{V}_l(x_1, x_2)}{\partial x_2} f_{2,l}(x_2, d_2) \right) \leq 0. \tag{4.7}$$

Let us assume that for each location $l \in L$, there exists a simulation function $\mathcal{V}_l$ of the continuous dynamics $C_{1,l}$ by $C_{2,l}$. We define the following sets which can be thought as some kind of neighborhoods associated with the simulation functions. For all $x_1 \in \mathbb{R}^{n_1}$, $\beta \geq 0$,

$$\mathcal{N}_l(x_1, \beta) = \{x_2 \in \mathbb{R}^{n_{2,l}} |\ \mathcal{V}_l(x_1, x_2) \leq \beta\}.$$

Then, it is possible to give the following characterization of approximate simulation relations for hybrid automata.

Theorem 4.2 (Approximate simulation relations for hybrid automata) *For all $l \in L$, let $\mathcal{V}_l$ be a simulation function of $C_{1,l}$ by $C_{2,l}$. Let $\{\beta_l |\ l \in L\}$ be positive numbers such that the following conditions hold:*

(a) for all $l \in L$, $\mathcal{N}_l(I_{1,l}, \beta_l) \subseteq I_{2,l}$,

(b) for all $e = (l, l') \in E$, $\mathcal{N}_l(G_{1,e}, \beta_l) \subseteq G_{2,e}$,

(c) for all $e = (l, l') \in E$,

$$\beta_{l'} \geq \sup_{\substack{x_1 \in G_{1,e} \\ \mathcal{V}_l(x_1, x_2) \leq \beta_l}} \left(\sup_{x_1' \in R_{1,e}(x_1)} \inf_{x_2' \in R_{2,e}(x_2)} \mathcal{V}_{l'}(x_1', x_2') \right).$$

(d) for all $l \in L$,

$$\beta_l \geq \sup_{x_1 \in I_{1,l}^0} \inf_{x_2 \in I_{2,l}^0} \mathcal{V}_l(x_1, x_2),$$

Let $\varepsilon = \max\{\beta_l |\ l \in L\}$. Then, the relation $\mathcal{R}_\varepsilon \subseteq X_1 \times X_2$ defined by

$$\mathcal{R}_\varepsilon = \{(l_1, x_1, l_2, x_2) |\ l_1 = l_2 = l,\ \mathcal{V}_l(x_1, x_2) \leq \beta_l\}$$

is an approximate simulation relation of $T(\Sigma_1)$ by $T(\Sigma_2)$ of precision ε and $T(\Sigma_1) \preceq_\varepsilon T(\Sigma_2)$.

The existence of simulation functions and assumption (a) guarantees that $T(\Sigma_2)$ can match the continuous transitions of $T(\Sigma_1)$. Assumptions (b) and (c) does the same for discrete transitions while assumption (d) takes care of the initial states.

It is clear that the positive numbers $\{\beta_l |\ l \in L\}$ cannot be chosen independently as they are linked by assumption (c) which can be interpreted as a condition of limitation of the expansion of the approximation error propagating through reset maps. Thus, it is not necessarily the case that there exist numbers such that assumptions of the theorem hold. In [GJP08], we identified several classes of hybrid automata for which we can guarantee their existence and define a procedure for their computation: acyclic hybrid automata, hybrid automata with memoryless or contracting resets.

Example 4.2 *We illustrate our approximation framework for hybrid systems in the context of a planar robot motion. Let us consider the hierarchical control architecture presented in Example 4.1 for the second order model of a robot:*

$$\begin{cases} \ddot{\mathbf{y}}_1(t) &= \mathbf{v}(t) - 1.0006(\mathbf{y}_1(t) - \mathbf{w}(t)) - 2.0005\dot{\mathbf{y}}_1(t) \\ \dot{\mathbf{w}}(t) &= \mathbf{v}(t) \end{cases}$$

where $\mathbf{y}_1(t) \in \mathbb{R}^2$ *denotes the position of the robot in a planar environment,* $\mathbf{w}(t) \in \mathbb{R}^2$ *is the internal variable of the controller and* $\mathbf{v}(t) \in \mathbb{R}^2$ *is the input of the hierarchical controller. We assume that initially* $\dot{\mathbf{y}}_1(0) = 0$ *and* $\mathbf{w}(0) = \mathbf{y}_1(0)$*. The value of the input* $\mathbf{v}(t) \in \{v_1, \ldots, v_6\}$ *(with* $\|v_1\| = \cdots = \|v_6\| = 0.2$*) is computed by a hybrid controller shown in the left part of Figure 4.3. This results in a hybrid automaton with* 6 *locations and* 6*-dimensional affine continuous dynamics. Let us remark that the reset maps of the hybrid automaton are equal to the identity.*

We have shown in Example 4.1 that the robot behaves approximately like the first order system

$$\dot{\mathbf{y}}_2(t) = \mathbf{v}(t).$$

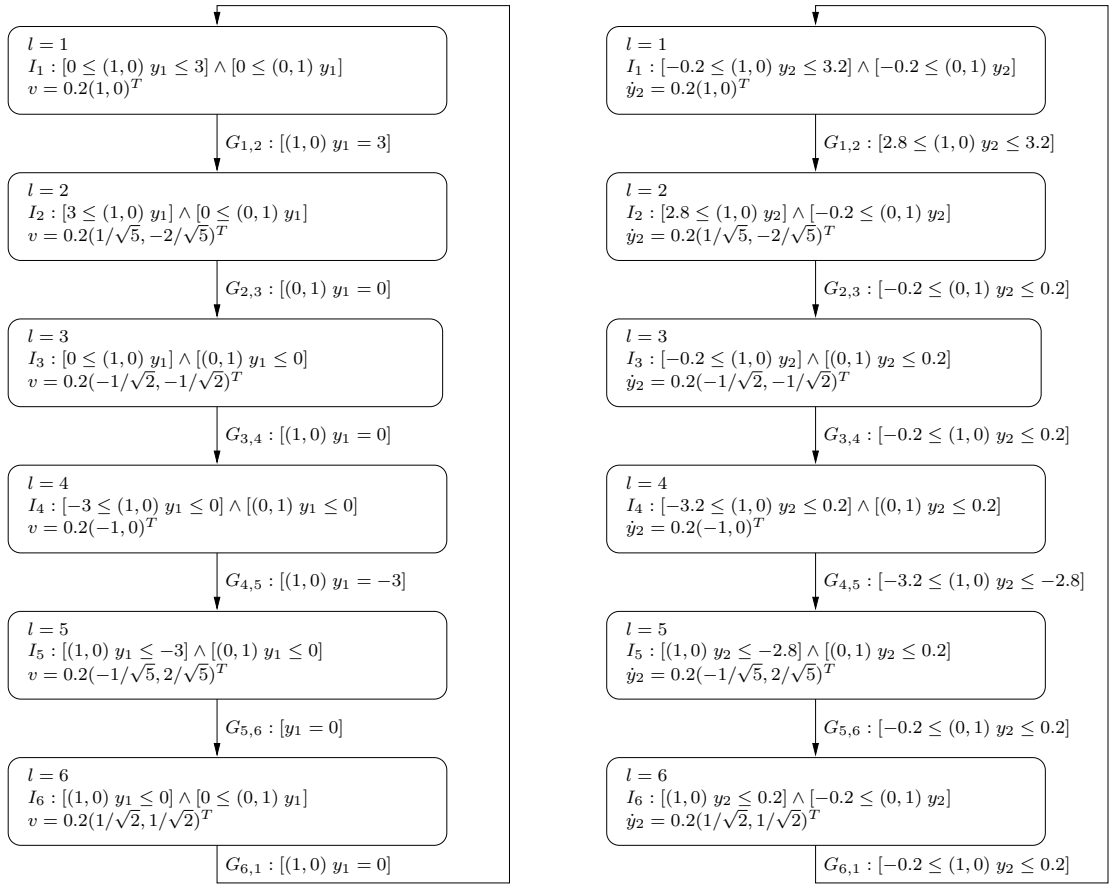

Figure 4.3: Hybrid controller for the mobile robot with hierarchical control architecture (left); Approximating hybrid automaton (right).

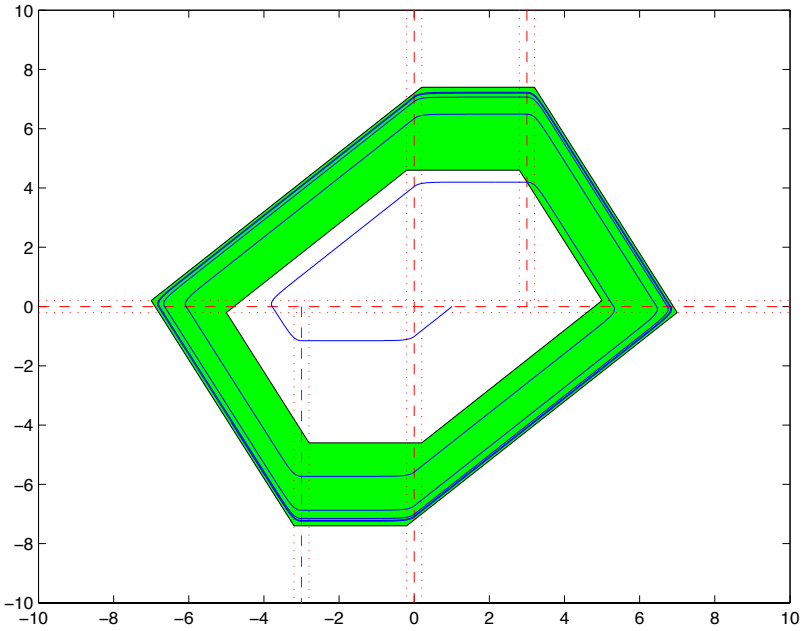

Figure 4.4: Trajectory of the mobile robot and attractor set of the approximating hybrid automaton. The dashed lines represent the guards of the original hybrid controller. The dotted lines represent the guards of the approximating hybrid automaton.

Hence, we shall approximate the hybrid automaton described above by another hybrid automaton with 6 *locations and* 2*-dimensional constant dynamics. We can check that the function*

$$\mathcal{V}(w, y_1, \dot{y}_1, y_2) = \max\left(\sqrt{\|w - y_1\|^2 + 8890.1\|w - y_1 - \dot{y}_1\|^2}, 0.2\right) + \|w - y_2\|$$

is a common simulation function for the continuous dynamics in each location. Then, it is easy to verify that the assumptions (c) and (d) of Theorem 4.2 hold with $\beta_1 = \cdots = \beta_6 = 0.2$*. We then choose the invariants and the guards so that assumptions (a) and (b) hold as well. The resulting approximating hybrid automaton is shown in the right part of Figure 4.3.*

It is easy to show that this hybrid automaton has a global attractor represented as the green region in Figure 4.4. Then, it follows that all the trajectories of the mobile robot will asymptotically be at distance at most 0.2 *of this attractor.*

4.3 Verification using Trajectory Simulation

The verification problem consists in analyzing the behavior of a dynamical system Σ against some specification: given a property Φ defined on trajectories (e.g. "a trajectory reaches the set Y_F"), we would like to be able to prove that it is satisfied by all trajectories of the system. For discrete systems, the problem has attracted a lot of attention in

computer science, resulting in the success story of Model Checking [CGP00] with the associated set of techniques widely used in the industry. Verification of continuous and hybrid systems is in general much more challenging since these generally have an uncountable number of trajectories. There has also been a lot of work on the subject using set-based reachability computations, abstraction and deductive techniques (see [Alu11] for a recent survey).

In the following, we briefly describe an approach presented in [FGP06] and based on simulation of individual trajectories of a system together with the use of auto-bisimulation function (that is bisimulation function between a system and itself). Let us consider the following dynamical system:

$$\Sigma : \left\{ \begin{array}{rcl} \dot{\mathbf{x}}(t) & = & f(\mathbf{x}(t)),\ \mathbf{x}(t) \in \mathbb{R}^n,\ \mathbf{x}(0) \in X^0 \\ \mathbf{y}(t) & = & g(\mathbf{x}(t)),\ \mathbf{y}(t) \in \mathbb{R}^k. \end{array} \right.$$

Let be given a property Φ defined on the output trajectories $\mathbf{y}$, formulated for instance in some temporal logic; $\mathbf{y} \models \Phi$ means that trajectory $\mathbf{y}$ satisfies property Φ. Let us assume that we are given a measure $\rho(\mathbf{y}, \Phi)$ estimating how robustly property Φ is satisfied by $\mathbf{y}'$:

$$\big(\forall t \geq 0,\ \|\mathbf{y}(\mathbf{t}) - \mathbf{y}'(t)\| \leq \rho(\mathbf{y}, \Phi)\big) \implies \mathbf{y}' \models \Phi.$$

We refer the reader to [FGP06] for a precise definition of such a measure and algorithms for its computation. The last ingredient of the verification approach is a bisimulation function between Σ and itself:

Definition 4.4 (Auto-bisimulation function) *Let $\mathcal{V} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ be a smooth function, $\mathcal{V}$ is an auto-bisimulation function of Σ if for all $(x_1, x_2) \in \mathbb{R}^n \times \mathbb{R}^n$,*

$$\mathcal{V}(x_1, x_2) \geq \|g(x_2) - g(x_1)\|$$

$$\frac{\partial \mathcal{V}(x_1, x_2)}{\partial x_1} \cdot f(x_1) + \frac{\partial \mathcal{V}(x_1, x_2)}{\partial x_2} \cdot f(x_2) \leq 0$$

Using robustness measures for property satisfaction and an auto-bisimulation function makes it possible to verify that the property holds for an infinite number of trajectories by simulating only one trajectory:

Proposition 4.2 *Let $x^0 \in X^0$ be an initial condition of Σ and let $\mathbf{y}$ be the associated output trajectory. Then, for all $x'^0 \in X^0$, with associated output trajectories $\mathbf{y}'$*

$$\big(\mathcal{V}(x^0, x'^0) \leq \rho(\mathbf{y}, \Phi)\big) \implies \mathbf{y}' \models \Phi$$

The previous result allows us to verify that the property Φ holds for all the trajectories of Σ by computing only a finite number of them. Let $\{x_1^0, \ldots, x_n^0\} \subseteq I^0$ be a finite subset of initial conditions and $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ the associated output trajectories of Σ such that for all $x^0 \in I^0$, there exists x_i^0 such that $\mathcal{V}(x^0, x_i^0) \leq \rho(\mathbf{y}_i, \Phi)$ then all the trajectories of Σ satisfy Φ. An algorithm to construct iteratively the sample of initial states can be found in [FGP06]. In the case we cannot cover the whole set of initial states, the algorithm identifies a subset of initial states for which the property holds. An interesting feature of the approach is that verification of properties that are robustly satisfied requires the simulation of a little number of trajectories.

Discussion: The results presented in Section 4.1 have been developed in collaboration with George J. Pappas, University of Pennsylvania. The proofs can be found in [GP09]. In collaboration with Georgios E. Fainekos, Arizona State University, and Hadas Kress Gazit, Cornell University, we have used this hierarchical control architecture, in combination with discrete abstractions techniques, to solve control problems in robotics with complex temporal logic specifications in [FGKGP09]. In collaboration with Alessandro Colombo, Politecnico di Milano, we have extended the approach developed for hierarchical control of linear systems to the class of differentially flat systems [CG13] (see also a similar work in [FST13]). An approach with similar flavors has been proposed in [CL07a, CL07b] for hierarchical stabilization and tracking control of linear systems. The hierarchical control methods presented in [Tab08, TI08], based on the use of discrete abstractions and approximate simulation, are also similar in spirit to the one presented here in the sense that some kind of interface is used to compute the inputs of the original model from those of the abstraction.

The approximation techniques for hybrid systems presented in Section 4.2 have been developed in collaboration with George J. Pappas and A. Agung Julius, Rensselaer Polytechnic Institute. The proofs can be found in [GJP08]. These techniques have been extended and used for verification of air-traffic coordination protocols [PVVD09] and cruise-controller implementations [QFD11b]. Similar approximation techniques have been developed for several classes of dynamical systems: linear systems with constrained inputs [GP07a], polynomial dynamical systems [GP05], stochastic hybrid systems for which a notion of stochastic bisimulation function is needed [JP09]. These approaches have been used for approximating the dynamics of complex biological [MIB+12], mechanical [TAJP08], or multi-agent robotic systems [MEB08].

The results of Section 4.3 on verification using trajectory simulation, developed with Georgios E. Fainekos and George J. Pappas have been published in [FGP06]. Similar approaches have been developed for the verification of systems with inputs [GP06], of hybrid systems [JFA+07] and of embedded control software [LKCK08]. These results were later improved in [GZ12] within the ANR Project VAL-AMS in collaboration with Gang Zheng during his postdoctoral stay at Laboratoire Jean Kuntzmann. The same kind of ideas can be used for controller synthesis by defining control laws from a finite number of trajectories [JA10].

Part II

Reachability Analysis

Chapter 5

Reachability Analysis of Continuous Systems

Résumé : *L'analyse d'atteignabilité est un sujet de recherche majeur dans le domaine des systèmes hybrides. Cette approche cherche à calculer l'ensemble des trajectoires d'un système dynamique, pour toutes les conditions initiales et sous toutes les perturbations ou variations de paramètres admissibles. Une analyse fructueuse par cette méthode permet de remplacer un nombre infini de simulations individuelles et d'obtenir des informations précieuses sur les propriétés du système considéré. En outre, les techniques de calcul d'ensembles atteignables sont utiles pour la synthèse de contrôleurs, la vérification ou le calcul d'abstractions symboliques. Dans ce chapitre, nous décrivons nos contributions à l'analyse d'atteignabilité des systèmes continus. Au-delà de l'intérêt intrinsèque de ce problème, il s'agit également d'une brique essentielle de toute méthode de calcul de l'ensemble des états atteignables d'un système hybride. Nous considérons d'abord les systèmes linéaires en temps continu avec entrées bornées. Pour cette classe de systèmes, nous proposons un schéma d'approximation permettant de calculer une sur-approximation de l'ensemble atteignable sur un intervalle de temps borné [LG10]. L'approximation est basée sur une discrétisation du temps et est donnée par l'union d'un nombre fini d'ensembles convexes. Elle peut être aussi précise que souhaitée en choisissant un pas de temps suffisamment petit. Nous proposons ensuite plusieurs implémentations de ce schéma d'approximation. La première est basée sur les zonotopes, une classe de polytopes, présentant des propriétés algorithmiques intéressantes [Gir05]. Nous décrivons une implémentation optimisée permettant un calcul efficace d'une approximation de l'ensemble atteignable [GLM06]. Nous étendons ensuite notre approche à des ensembles convexes arbitraires en proposant une implémentation basée sur l'utilisation des fonctions support [LG10]. Dans la deuxième partie de ce chapitre, nous considérons des systèmes dynamiques polynomiaux en temps discret. Nous présentons une approche pour calculer des sur-approximations polyédriques de l'ensemble atteignable en utilisant des relaxations de problèmes d'optimisation polynomiale à l'aide de programmes linéaires obtenus grâce aux propriétés de la forme de Bernstein des polynômes [BTDG12].*

Hybrid systems research explores models that combine discrete and continuous dynamics, and attempts to extend specific analysis methods developed for each type of systems toward methods that can analyze the behavior of a complete system, having both types of dynamics. An approach that has emerged from this area consists of a combination of ideas from algorithmic verification of discrete systems (Model Checking [CGP00]) and numerical simulation of continuous systems. This approach seeks to compute (an approximation of) the set of all trajectories of the system, starting from all possible initial conditions, and under all admissible disturbances and variations in parameter values. A successful analysis according to this method can replace infinitely many individual simulations and give additional insight on the properties of the system under study. One can view this approach as a compromise between analytical methods that give strong results but apply mostly to fairly simple systems, and simulation-based methods that can be applied, in principle, to arbitrary classes of systems, but whose results cannot guarantee absolute confidence. Moreover, computational techniques for reachability analysis have been shown to be powerful tools for several problems related to analysis and control of hybrid systems such as controller synthesis, verification or computation of symbolic abstractions [ABD+00, TMBO03, ADI06]. For these approaches to be mathematically sound, we often need to guarantee some properties of the computed approximation. For instance, it is often necessary that the computed approximation includes the true reachable set. Hence, most of the existing work seek to compute over-approximations of the reachable set.

Computing reachable states for continuous or hybrid systems has become a major research issue in hybrid systems. For hybrid systems in which the continuous dynamics is a constant differential inclusion in each location, such as timed automata or "linear" hybrid automata, the exact computation of the reachable states can be done with elementary manipulation of polytopes [ACH+95, AMP95, HHWT97, Fre08]. For systems with linear continuous dynamics, an approximation of the reachable states is generally computed by a combination of numerical integration and geometrical algorithms on polytopes [GM99, CK99, ABDM00, SK03, Gir05, GLM06, HK06, LG09, LG10, ASB10, FLD+11] or ellipsoids [KV00, BT00, KV07]. Nonlinear continuous dynamics are much more difficult to handle and computation of their reachable states can be done by solving a partial differential equation [MT00], by extending the linear systems reachability analysis using local linear approximations of the dynamics [ADG03, ASB08, DMT10] or by exploiting properties of specific classes of nonlinear systems such as polynomial or monotone systems [Dan06, DS09, RMC10, BTDG12].

In this chapter, we review our contributions to reachability analysis of continuous systems. These techniques constitutes an essential ingredient of algorithms for computing the set of reachable states of hybrid systems. We first consider continuous-time linear systems with compact convex sets of inputs. For this class of systems, we propose an approximation scheme that allows us to compute an over-approximation of the reachable set on a bounded time interval. The approximation is based on time-discretization and is given by the union of a finite number of convex sets. It can be made arbitrarily accurate by choosing a time step small enough. We then propose several implementations of this approximation scheme. The first one is based on zonotopes, a class of polytopes with interesting computational properties. We propose an improved implementation that allows us to compute efficiently an approximation of the reachable set. We then extend our approach to arbitrary convex sets by proposing an implementation based on support

functions. In the second part of the chapter, we consider discrete-time polynomial dynamical systems and propose an approach to compute polytopic over-approximations of the reachable set using linear programming relaxations of polynomial optimization problems based on properties of the Bernstein form of polynomials.

5.1 Reachability Analysis of Linear Systems

We consider continuous-time linear systems of the form:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t),\ \mathbf{x}(t) \in \mathbb{R}^n,\ \mathbf{u}(t) \in U$$

where $U \subseteq \mathbb{R}^k$ is a compact convex set, A and B are matrices of compatible dimensions. Given a subset $X \subseteq \mathbb{R}^n$, we denote by $\mathcal{R}_s(X) \subseteq \mathbb{R}^n$ the set of states reachable by the system at time s from states in X:

$$\mathcal{R}_s(X) = \{\mathbf{x}(s) |\ \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t), \mathbf{u}(t) \in U, \forall t \in [0, s] \text{ and } \mathbf{x}(0) \in X\}.$$

Then, the reachable set on the time interval $[s, s']$ is defined as

$$\mathcal{R}_{[s,s']}(X) = \bigcup_{t \in [s,s']} \mathcal{R}_t(X).$$

Let $X_0 \subseteq \mathbb{R}^n$ be a specified compact convex set of initial states. In the following, we are interested in computing an over-approximation of the reachable set on the time interval $[0, T]$ from the initial states X_0, that is $\mathcal{R}_{[0,T]}(X_0)$.

Let $V = BU \subseteq \mathbb{R}^n$, let $\|.\|$ be a norm on $\mathbb{R}^n$, $\|A\|$ is the induced norm of the matrix A. We shall denote

$$R_{X_0} = \max_{x \in X_0} \|x\|,\ D_{X_0} = \max_{x,y \in X_0} \|x - y\|,\ \text{and}\ R_V = \max_{v \in V} \|v\|.$$

We define the following elementary operations on sets. Given a set $\Omega \subseteq \mathbb{R}^n$ and a matrix A, $A\Omega$ denotes the image of Ω by A. Given a real number λ, $\lambda\Omega = (\lambda I)\Omega$ where I is the identity matrix. Let Ω, $\Omega' \subseteq \mathbb{R}^n$, $\mathsf{Conv}(\Omega, \Omega')$ denotes the convex hull of Ω and Ω' and $\Omega \oplus \Omega'$ denotes the Minkowski sum of Ω and Ω': $\Omega \oplus \Omega' = \{x + x' :\ x \in \Omega, x' \in \Omega'\}$. We shall also consider two notions of approximation of sets based on two distances. Given Ω and Ω' two compact subsets of $\mathbb{R}^d$, we define the distance between Ω and Ω':

$$d(\Omega, \Omega') = \inf_{x \in \Omega} \inf_{x' \in \Omega'} \|x - x'\|.$$

We also define the Hausdorff distance between Ω and Ω':

$$d_H(\Omega, \Omega') = \max\left(\sup_{x \in \Omega} \inf_{x' \in \Omega'} \|x - x'\|, \sup_{x' \in \Omega'} \inf_{x \in \Omega} \|x - x'\|\right).$$

Let us remark that only the Hausdorff distance is a metric in the usual sense. Particularly, $d_H(\Omega, \Omega') = 0$ if and only if $\Omega = \Omega'$ whereas $d(\Omega, \Omega') = 0$ if and only if $\Omega \cap \Omega'$ is not empty.

5.1.1 Time-discretization scheme

In this section, we show how the reachable set can be over-approximated by the union of convex sets [LG10]. Further, the Hausdorff distance between the reachable set and its approximation can be made arbitrarily small. Our approach for approximating the reachable set is based on a discretization of the time. Let $\tau = T/N$ be the time step (with $N \in \mathbb{N}$). Then, we have:

$$\mathcal{R}_{[0,T]}(X_0) = \bigcup_{i=0}^{N-1} \mathcal{R}_{[i\tau,(i+1)\tau]}(X_0).$$

In order to compute an over-approximation of $\mathcal{R}_{[0,T]}(X_0)$, we shall compute over-approximations of all the sets $\mathcal{R}_{[i\tau,(i+1)\tau]}(X_0)$. We consider the first element of the sequence, $\mathcal{R}_{[0,\tau]}(X_0)$:

Lemma 5.1 *Let $\Omega_0 \subseteq \mathbb{R}^n$ be the convex set defined by:*

$$\Omega_0 = \mathsf{Conv}\left(X_0, e^{\tau A}X_0 \oplus \tau V \oplus \alpha_\tau \mathbb{B}\right) \tag{5.1}$$

where $\alpha_\tau = (e^{\tau\|A\|} - 1 - \tau\|A\|)(R_{X_0} + \frac{R_V}{\|A\|})$ and $\mathbb{B}$ denotes the unit ball for the considered norm. Then, $\mathcal{R}_{[0,\tau]}(X_0) \subseteq \Omega_0$ and

$$d_H(\Omega_0, \mathcal{R}_{[0,\tau]}(X_0)) \leq \frac{1}{4}(e^{\tau\|A\|} - 1)D_{X_0} + 2\alpha_\tau.$$

This lemma can be roughly understood as follows, $e^{\tau A}X_0 \oplus \tau V$ is an approximation of the reachable set at time τ; the convex hull of X_0 and $e^{\tau A}X_0 \oplus \tau V$ gives an approximation of $\mathcal{R}_{[0,\tau]}(X_0)$. The role of the bloating factor α_τ is to ensure over-approximation. The approximation error can be made arbitrarily small by choosing τ small enough. We now consider the other elements of the sequence $\mathcal{R}_{[i\tau,(i+1)\tau]}(X_0)$. Let us remark that we have

$$\mathcal{R}_{[(i+1)\tau,(i+2)\tau]}(X_0) = \mathcal{R}_\tau\left(\mathcal{R}_{[i\tau,(i+1)\tau]}(X_0)\right),\ i = 0, \ldots, N-2.$$

Given $\Omega \subseteq \mathbb{R}^n$, the following lemma provides us with an over-approximation of $\mathcal{R}_\tau(\Omega)$:

Lemma 5.2 *Let $\Omega \subseteq \mathbb{R}^n$, let $\Omega' \subseteq \mathbb{R}^n$ be the set defined by:*

$$\Omega' = e^{\tau A}\Omega \oplus \tau V \oplus \beta_\tau \mathbb{B}$$

where $\beta_\tau = (e^{\tau\|A\|} - 1 - \tau\|A\|)\frac{R_V}{\|A\|}$ and $\mathbb{B}$ denotes the unit ball for the considered norm. Then, $\mathcal{R}_\tau(\Omega) \subseteq \Omega'$ and

$$d_H(\Omega', \mathcal{R}_\tau(\Omega)) \leq 2\beta_\tau.$$

The set $e^{\tau A}\Omega \oplus \tau V$ is an approximation of the reachable set at time τ; bloating this set using the ball of radius β_τ ensures over-approximation. Again, the approximation error can be made arbitrarily small by choosing τ small enough. We shall now define the sequence of convex sets Ω_i over-approximating $\mathcal{R}_{[i\tau,(i+1)\tau]}(X_0)$ as follows. Ω_0 is given by equation (5.1) and

$$\Omega_{i+1} = e^{\tau A}\Omega_i \oplus \tau V \oplus \beta_\tau \mathbb{B},\ i = 0, \ldots, N-2. \tag{5.2}$$

Theorem 5.1 (Approximation of the reachable set) *Let the sets Ω_i be defined by equations (5.1) and (5.2); then, for all $i = 0, \ldots, N-1$, $\mathcal{R}_{[i\tau,(i+1)\tau]}(X_0) \subseteq \Omega_i$ and*

$$d_H(\Omega_i, \mathcal{R}_{[i\tau,(i+1)\tau]}(X_0)) \leq \tau e^{T\|A\|} \left(\frac{\|A\|}{4} D_{X_0} + \tau \|A\|^2 R_{X_0} + e^{\tau\|A\|} R_V \right).$$

This theorem essentially states that the reachable set $\mathcal{R}_{[0,T]}(X_0)$ can be over-approximated by the union of convex sets $\Omega_0 \cup \cdots \cup \Omega_{N-1}$. Further, the error bound for the Hausdorff distance is in $\mathcal{O}(\tau)$ and thus can be made arbitrarily small by choosing τ small enough.

There are numerous approaches implementing time-discretization schemes of the previous type for computing approximations of the reachable set. These mainly differ by the class of convex sets considered to represent the convex sets $\Omega_0, \ldots, \Omega_{N-1}$. These can be general polytopes [GM99, CK99, ABDM00], ellipsoids [KV00, BT00] or rectangles [SK03]. Polytopes are closed under linear transformations and Minkowski sum, so they are suitable to implement the recurrence relation (5.2). However, for polytopes, Minkowski sum is costly and the resulting polytope can be much more complex than the original polytopes. Thus, their use generally leads to precise but costly reachability computations and is limited to systems of modest dimension. Ellipsoids and rectangles result in efficient implementations, but these classes of convex sets are not closed under Minkowski sum and thus additional approximations are needed to implement the recurrence relation (5.2). In the following, we present implementations of the time-discretization scheme described above based on representation of convex sets using zonotopes and support functions.

5.1.2 Approximation using zonotopes

Zonotopes: Zonotopes form a special class of convex polytopes. Traditionally, a zonotope is defined as the image of a cube under an affine projection [Zie95]. Equivalently, a zonotope is a Minkowski sum of a finite set of line segments:

Definition 5.1 (Zonotope) *A zonotope Z in $\mathbb{R}^n$ is a set such that:*

$$Z = \left\{ x \in \mathbb{R}^n | \; x = c + \sum_{i=1}^{i=p} x_i g_i, \; -1 \leq x_i \leq 1 \right\}$$

where c, $g_1, \ldots, g_p$ are vectors of $\mathbb{R}^n$. We note $Z = (c, \langle g_1, \ldots, g_p \rangle)$.

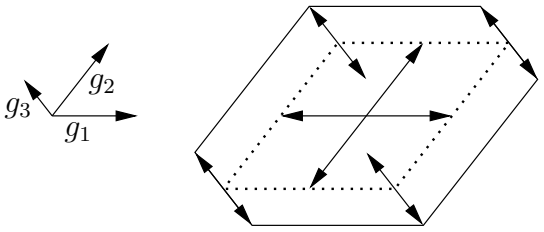

Figure 5.1: Example of a zonotope with three generators

It is clear that a zonotope is a polytope. Parallelepipeds and hyperrectangles are particular zonotopes. Note that a zonotope is always centrally symmetric. The point $c \in \mathbb{R}^n$ is called the *center* of Z. The vectors $g_1, \ldots, g_p$ are called the *generators* of Z. In Figure 5.1, we represented a planar zonotope with three generators. From a practical point of view, Definition 5.1 gives a compact representation of the set. Practical applications of zonotopes for rigorous approximation of dynamical systems have been shown in [Küh98, Com03, ABC05]. In the following, we use zonotopes to over-approximate the reachable set of a linear system. The use of zonotopes is motivated by two main properties:

Proposition 5.1 *Let $Z = (c, \langle g_1, \ldots, g_p \rangle)$ and $Z' = (c', \langle g'_1, \ldots, g'_q \rangle)$ be two zonotopes in $\mathbb{R}^n$, let A be a matrix with n columns, then*

1. *AZ is a zonotope: $AZ = (Ac, \langle Ag_1, \ldots, Ag_p \rangle)$;*
2. *$Z \oplus Z'$ is a zonotope: $Z \oplus Z' = (c + c', \langle g_1, \ldots, g_p, g'_1, \ldots, g'_q \rangle)$.*

Hence, zonotopes are closed under linear transformation and Minkowski sum and the computation of these operations is very efficient. The image of a zonotope by a linear map can be computed in linear time with regard to the number of generators while the Minkowski sum is computed in constant time by the concatenation of the lists of generators. Linear transformation and Minkowski sum are the elementary operations of the recurrence relation (5.2) which motivates the use of zonotopes for approximating the reachable set of a linear system.

Reachability analysis using zonotopes: In this section, we present our results for reachability analysis of linear systems using zonotopes [Gir05]. Let us assume that the set of initial states X_0 and inputs U of the linear system are specified by zonotopes. Then, $V = BU$ is a zonotope as well. Let $\|.\|$ denote the infinity norm on $\mathbb{R}^n$, then $\mathbb{B}$ is the unit cube which is a zonotope. Hence, the set $W_\tau = \tau V \oplus \beta_\tau \mathbb{B}$ is a zonotope as well. If Ω_0 is a zonotope, then all the sets $\Omega_1, \ldots, \Omega_{N-1}$ defined by the recurrence relation (5.2) are zonotopes. Unfortunately, the convex hull of two zonotopes is generally not a zonotope and Ω_0 defined by equation (5.1) is unlikely to be a zonotope. Hence, we shall initialize the sequence with a zonotope $\overline{\Omega}_0$ which contains Ω_0. Let $X_0 = (c, \langle g_1, \ldots, g_p \rangle)$, let us define

$$\overline{\Omega}_0 = (\tfrac{c+e^{\tau A}c}{2}, \langle \tfrac{g_1+e^{\tau A}g_1}{2}, \ldots, \tfrac{g_p+e^{\tau A}g_p}{2}, \tfrac{c-e^{\tau A}c}{2}, \tfrac{g_1-e^{\tau A}g_1}{2}, \ldots, \tfrac{g_p-e^{\tau A}g_p}{2} \rangle) \oplus \tau V \oplus \alpha_\tau \mathbb{B}. \tag{5.3}$$

Then, $\Omega_0 \subseteq \overline{\Omega}_0$. Let us consider Algorithm 1 which allows us to compute an over-approximation of the reachable set. Moreover, the following result shows that the error bound for the Hausdorff distance is in $\mathcal{O}(\tau)$ and thus can be made arbitrarily small by choosing τ small enough.

Proposition 5.2 *Let $\overline{\Omega}_0, \ldots, \overline{\Omega}_{N-1}$ be computed by Algorithm 1; for all $i = 0, \ldots, N-1$, $\mathcal{R}_{[i\tau,(i+1)\tau]}(X_0) \subseteq \overline{\Omega}_i$ and*

$$d_H(\overline{\Omega}_i, \mathcal{R}_{[i\tau,(i+1)\tau]}(X_0)) \leq \tau e^{T\|A\|} \left(\frac{\|A\|}{2} R_{X_0} + \tau \|A\|^2 R_{X_0} + 2R_V \right).$$

Algorithm 1 Over-approximation of the reachable set by zonotopes.

Require: The matrix $e^{\tau A}$, the zonotopes X_0 and W_τ, and an integer N.
Ensure: Zonotopes $\overline{\Omega}_i$ for i in $\{0, \ldots, N-1\}$ over-approximating the reachable set $\mathcal{R}_{[0,T]}(X_0)$
1: Let $\overline{\Omega}_0$ defined by equation (5.3).
2: **for** i from 0 to $N-2$ **do**
3: $\quad \overline{\Omega}_{i+1} \leftarrow e^{\tau A}\overline{\Omega}_i \oplus W_\tau$
4: **end for**
5: **return** $\{\overline{\Omega}_0, \ldots, \overline{\Omega}_{N-1}\}$

At each iteration of the loop of Algorithm 1, the zonotope $\overline{\Omega}_{i+1}$ is obtained by computing the image of $\overline{\Omega}_i$ by a linear map and by adding the zonotope W_τ. Consequently, the number of generators of the zonotope $\overline{\Omega}_i$ is proportional to i. Then, the memory needed to store the over-approximation of the set $\mathcal{R}_{[0,T]}(X_0)$ and the time needed for the computation are quadratic in the number of steps N. For large values of N the over-approximation of $\mathcal{R}_{[0,T]}(X_0)$ can thus be quite expensive in memory and in time.

A solution to avoid this quadratic cost is to limit the number of generators of the zonotopes $\overline{\Omega}_i$. Let $m \geq n$ be the maximum number of generators allowed for the zonotopes $\overline{\Omega}_i$. If the number of generators of the zonotope $e^{\tau A}\overline{\Omega}_i \oplus W_\tau$ is $m + m' > m$, then following Algorithm 1, the zonotope $\overline{\Omega}_{i+1}$ should have $m + m'$ generators which is greater than the maximum number allowed. We propose a supplementary approximation step in order to reduce the numbers of generators. It consists in taking $m' + n$ generators of $e^{\tau A}\overline{\Omega}_i \oplus W_\tau$, $h_1, \ldots, h_{m'+n}$, and to replace them by n generators, such that the new zonotope $\overline{\Omega}_{i+1}$ with m generators contains $e^{\tau A}\overline{\Omega}_i \oplus W_\tau$. Equivalently, we have to over-approximate the zonotope $(0, \langle h_1, \ldots, h_{m'+n}\rangle)$ by a zonotope with n generators. It is easy to verify that one can choose the product of intervals:

$$\left[-\sum_{i=1}^{i=m'+n} |h_{i,1}|\,,\ \sum_{i=1}^{i=m'+n} |h_{i,1}|\right] \times \cdots \times \left[-\sum_{i=1}^{i=m'+n} |h_{i,n}|\,,\ \sum_{i=1}^{i=m'+n} |h_{i,n}|\right]$$

which is a zonotope with n generators.

The choice of the $m' + n$ generators of $e^{\tau A}\overline{\Omega}_i \oplus W_\tau$ to be replaced is important for the quality of the approximation. The best selection consists in taking the vectors $h_1, \ldots, h_{m'+n}$ such that the over-approximation of the zonotope $(0, < h_1, \ldots, h_{m'+n} >)$ by a product of intervals is as good as possible (products of intervals are zonotopes whose generators have only one non zero component). Let $e^{\tau A}\overline{\Omega}_i \oplus W_\tau = (c, \langle g_1, \ldots, g_{m+m'}\rangle)$ and let us assume that the generators have been sorted such that:

$$\|g_1\|_1 - \|g_1\|_\infty \leq \|g_2\|_1 - \|g_2\|_\infty \leq \cdots \leq \|g_{m+m'}\|_1 - \|g_{m+m'}\|_\infty.$$

We choose for $i \in \{1, \ldots, m' + n\}$, $h_i = g_i$. These vectors are closed to vectors with only one non zero component and therefore $(0, < h_1, \ldots, h_{m'+n} >)$ is well approximated by a product of intervals.

Using this reduction step in Algorithm 1 allows us to compute an over-approximation of the reachable set $\mathcal{R}_{[0,T]}(X_0)$ in linear time and space. However, we loose the convergence with respect to the Haussdorf distance when the time step approaches zero. Even worse, when the number of steps N is large, the numerous successive approximations caused

by the reduction steps may propagate and result in a dramatic over-approximation error (phenomenom known as the *wrapping effect*) rendering the reachability analysis useless.

Improved algorithmic scheme: In this section, we describe an improved algorithmic scheme, developed in [GLM06], for computing the sequence of zonotopes $\overline{\Omega}_0, \ldots, \overline{\Omega}_{N-1}$ in linear time and space. Let us remark that for $i = 0, \ldots, N-2$ the set $\overline{\Omega}_{i+1}$ computed by Algorithm 1 satisfies

$$\overline{\Omega}_{i+1} = (e^{\tau A})^{i+1}\overline{\Omega}_0 \oplus (e^{\tau A})^i W_\tau \oplus \cdots \oplus W_\tau.$$

Then, let us define the auxiliary sequences:

$$\begin{array}{lllllll} Z_0 & = & \overline{\Omega}_0, & Z_{i+1} & = & e^{\tau A} Z_i, \\ W_0 & = & W_\tau, & W_{i+1} & = & e^{\tau A} W_i, \\ S_0 & = & \{0\}, & S_{i+1} & = & S_i \oplus W_i. \end{array} \tag{5.4}$$

Equivalently, we have

$$X_{i+1} = (e^{\tau A})^{i+1}\overline{\Omega}_0,\ W_{i+1} = (e^{\tau A})^{i+1} W_\tau \text{ and } S_{i+1} = (e^{\tau A})^i W_\tau \oplus \ldots \oplus W_\tau.$$

Therefore, $\overline{\Omega}_{i+1} = Z_{i+1} \oplus S_{i+1}$. Algorithm 2 implements the reachable set computation based on the recurrence relations (5.4).

Algorithm 2 Efficient over-approximation of the reachable set by zonotopes.

Require: The matrix $e^{\tau A}$, the zonotopes X_0 and W_τ, and an integer N.
Ensure: Zonotopes $\overline{\Omega}_i$ for i in $\{0, \ldots, N-1\}$ over-approximating the reachable set $\mathcal{R}_{[0,T]}(X_0)$

1: Let $\overline{\Omega}_0$ defined by equation (5.3).
2: $Z_0 \leftarrow \overline{\Omega}_0$
3: $W_0 \leftarrow W_\tau$
4: $S_0 \leftarrow \{0\}$
5: **for** i from 0 to $N-2$ **do**
6: $\quad Z_{i+1} \leftarrow e^{\tau A} Z_i$
7: $\quad W_{i+1} \leftarrow e^{\tau A} W_i$
8: $\quad S_{i+1} \leftarrow S_i \oplus W_i$
9: $\quad \overline{\Omega}_{i+1} \leftarrow Z_{i+1} \oplus S_{i+1}$
10: **end for**
11: **return** $\{\overline{\Omega}_0, \ldots, \overline{\Omega}_{N-1}\}$

The main advantage of this algorithm is that the linear transformations are applied to the sets Z_i and W_i whose complexity does not increase at each iteration and this constitutes a significant improvement over Algorithm 1. This major improvement results in a linear time complexity for Algorithm 2, since Minkowski sums are computed in constant time for zonotopes. Regarding the space complexity, since the Minkowski sum essentially consists of a concatenation of lists, it is not necessary to store the sequence S_i and $\overline{\Omega}_i$ since it can be computed very easily from the sequences Z_i and W_i. Therefore, the space complexity of Algorithm 2 is linear in the number of steps N.

Algorithm 2 hence allows us to approximate the reachable set efficiently without approximations other than those implied by the use of the recurrence relation (5.2) and

the approximation of the initial set Ω_0 by equation (5.3). Let us remark that the sets $\overline{\Omega}_0, \ldots, \overline{\Omega}_{N-1}$ computed by Algorithm 2 are exactly the same as those computed by Algorithm 1. Therefore, the approximation result given by Proposition 5.2 remains valid and the wrapping effect is avoided.

5.1.3 Approximation using support functions

In this section, we present a second implementation of the time-discretization scheme presented in Section 5.1.1 that can handle arbitrary compact convex sets by using the notion of support function. These results have been presented in [LG10].

Convex sets and support functions: The support function of a convex set is a classical tool of convex analysis. In the following, we shall use support functions as a representation of arbitrary compact convex sets.

Definition 5.2 (Support function) *Let $\Omega \subseteq \mathbb{R}^n$ be a compact convex set; the support function of Ω, denoted ρ_Ω, is defined as:*

$$\begin{array}{rccl} \rho_\Omega : & \mathbb{R}^n & \to & \mathbb{R} \\ & \ell & \mapsto & \max_{x \in \Omega} \ell \cdot x \end{array}$$

The notion of support function is illustrated in Figure 5.2. It can be shown that the support function of a compact convex set is a convex function.

It is to be noted that the set Ω is uniquely determined by its support function as the following equality holds:

$$\Omega = \bigcap_{\ell \in \mathbb{R}^n} \{x \in \mathbb{R}^n : \ell \cdot x \leq \rho_\Omega(\ell)\} \tag{5.5}$$

which means that any convex set Ω is the intersection of the infinite set of halfspaces with normal vector $\ell \in \mathbb{R}^n$ and distance value $\rho_\Omega(\ell)$.

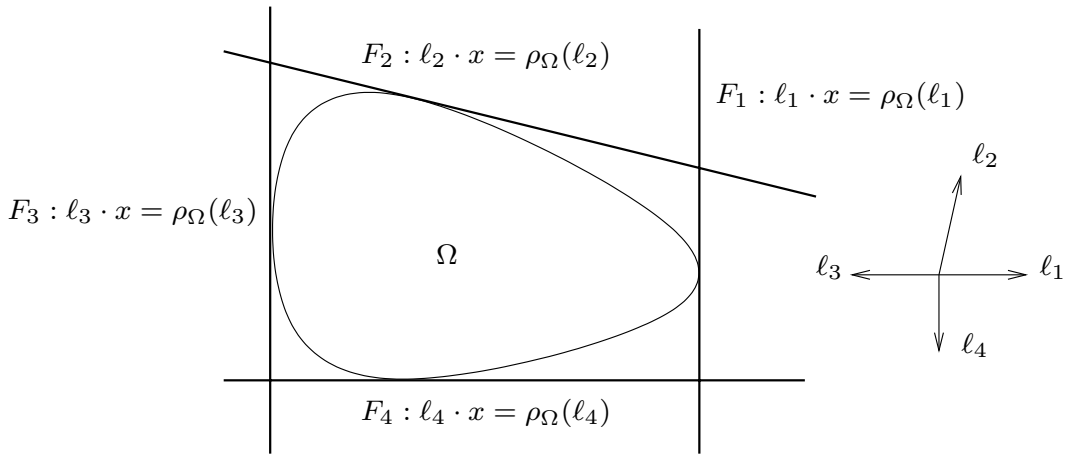

Figure 5.2: Illustration of the notion of support function of a convex set Ω.

Proposition 5.3 *For the following compact convex sets, the support function can be computed.*

- *An ellipsoid:* $\Omega = \left\{x \in \mathbb{R}^n |\ x^\top Q^{-1} x \leq 1\right\}$ *where* Q *is a positive definite symmetric matrix. Then,*
$$\rho_\Omega(\ell) = \sqrt{\ell^\top Q \ell}.$$

- *A zonotope:* $\Omega = \{\alpha_1 g_1 + \cdots + \alpha_r g_r |\ \alpha_j \in [-1, 1],\ j = 1, \ldots, r\}$ *where the generators* $g_1, \ldots, g_r \in \mathbb{R}^n$*. Then,*
$$\rho_\Omega(\ell) = \sum_{j=1}^{r} |g_j \cdot \ell|.$$

- *A polytope:* $\Omega = \{x \in R^n |\ Cx \leq d\}$ *where* C *and* d *are a matrix and vector of compatible dimension. Then, computing* $\rho_\Omega(\ell)$ *is equivalent to solving the linear program:*
$$\begin{cases} \textit{Maximize } \ell \cdot x \\ \textit{Subject to } Cx \leq d \end{cases}$$

Further, more complex sets can be considered using operations on elementary convex sets. The support function of sets defined using these operations can be computed using the following properties.

Proposition 5.4 *For all compact convex sets* $\Omega,\ \Omega' \subseteq \mathbb{R}^n$*, for all matrices* A*, all real numbers* λ*, and all vectors* $\ell \in \mathbb{R}^n$*, we have:*

$$\begin{aligned}
\rho_{A\Omega}(\ell) &= \rho_\Omega(A^\top \ell) \\
\rho_{\lambda\Omega}(\ell) &= \rho_\Omega(\lambda\ell) \ = \ \lambda\rho_\Omega(\ell) \\
\rho_{\mathsf{Conv}(\Omega,\Omega')}(\ell) &= \max(\rho_\Omega(\ell), \rho_{\Omega'}(\ell)) \\
\rho_{\Omega\oplus\Omega'}(\ell) &= \rho_\Omega(\ell) + \rho_{\Omega'}(\ell). \\
\rho_{\Omega\cap\Omega'}(\ell) &= \inf_{\omega\in\mathbb{R}^n} (\rho_\Omega(\ell - \omega) + \rho_{\Omega'}(\omega)) \leq \min(\rho_\Omega(\ell), \rho_{\Omega'}(\ell)).
\end{aligned}$$

From equation (5.5), it is easy to see that polytopic over-approximations of an arbitrary compact convex set can be obtained by "sampling" its support function.

Proposition 5.5 *Let* Ω *be a compact convex set and* $\ell_1, \ldots, \ell_r \in \mathbb{R}^n$ *be arbitrarily chosen vectors; let us define the following polytope:*

$$\overline{\Omega} = \{x \in \mathbb{R}^n |\ \ell_k \cdot x \leq \rho_\Omega(\ell_k), k = 1, \ldots, r\}.$$

Then, $\Omega \subseteq \overline{\Omega}$*. Moreover, we say that this over-approximation is tight as* Ω *touches the faces* $F_1, \ldots, F_r$ *of* $\overline{\Omega}$*:*
$$d(\Omega, F_k) = 0,\ k = 1, \ldots, r.$$

An example of such polytopic over-approximation of a convex set can be seen in Figure 5.2.

Reachability analysis using support functions: We now consider the computation of the support functions of the approximate reachable sets $\Omega_0 \dots \Omega_{N-1}$ defined by the time-discretization scheme presented in Section 5.1.1. For simplicity of the notations, let us introduce the matrix Φ_τ and the set W_τ defined by

$$\Phi_\tau = e^{\tau A},\ W_\tau = \tau V \oplus \beta_\tau \mathbb{B}. \tag{5.6}$$

Using Proposition 5.4, it follows that the support function of W_τ is given by

$$\rho_{W_\tau}(\ell) = \tau \rho_V(\ell) + \beta_\tau \rho_{\mathbb{B}}(\ell) \tag{5.7}$$

where $\rho_{\mathbb{B}}$ is the support function of the unit ball for the chosen norm. The following proposition gives the expression of $\rho_{\Omega_0} \dots \rho_{\Omega_{N-1}}$.

Proposition 5.6 *Let $\Omega_0 \dots \Omega_{N-1}$ be the sets defined by equations (5.1) and (5.2). Then, for all ℓ in $\mathbb{R}^n$,*

$$\rho_{\Omega_0}(\ell) = \max\left(\rho_{X_0}(\ell), \rho_{X_0}(\Phi_\tau^\top \ell) + \tau \rho_V(\ell) + \alpha_\tau \rho_{\mathbb{B}}(\ell)\right) \tag{5.8}$$

and for $i = 0, \dots, N-1$,

$$\rho_{\Omega_i}(\ell) = \rho_{\Omega_0}\left((\Phi_\tau^\top)^i \ell\right) + \sum_{j=0}^{i-1} \rho_{W_\tau}\left((\Phi_\tau^\top)^j \ell\right).$$

Hence, we showed that the reachable set of a linear system can be over-approximated arbitrarily close by a union of compact convex sets with effectively computable support functions. The representation of convex sets by their support function is not suitable for some tasks, especially when an explicit representation is needed. From Proposition 5.5, polytopic approximations of the sets $\Omega_0, \dots, \Omega_{N-1}$ can be obtained by evaluating their support functions in several directions. These sets provide with polytopic approximations of the reachable sets:

Proposition 5.7 *Let $\rho_{\Omega_0}, \dots, \rho_{\Omega_{N-1}}$ be the functions defined in Proposition 5.6. Let $\ell_1, \dots, \ell_r \in \mathbb{R}^n$ be arbitrarily chosen vectors; let us define the following polytope:*

$$\overline{\Omega}_i = \{x \in \mathbb{R}^n |\ \ell_k \cdot x \leq \rho_{\Omega_i}(\ell_k), k = 1, \dots, r\},\ i = 0, \dots, N-1.$$

Then, for all $i = 0, \dots, N-1$, $\mathcal{R}_{[i\tau,(i+1)\tau]}(X_0) \subseteq \overline{\Omega}_i$. Let $F_{i,1}, \dots, F_{i,r}$ denote the faces of polytope $\overline{\Omega}_i$, then

$$d(\mathcal{R}_{[i\tau,(i+1)\tau]}(X_0), F_{i,k}) \leq \tau e^{T\|A\|}\left(\frac{\|A\|}{4} D_{X_0} + \tau \|A\|^2 R_{X_0} + e^{\tau\|A\|} R_V\right).$$

Proposition 5.7 states that by evaluating the functions $\rho_{\Omega_0}, \dots, \rho_{\Omega_{N-1}}$, we can compute a union of polytopes over-approximating the reachable set $\mathcal{R}_{[0,T]}(X_0)$. Moreover, the distance between each face of the approximating polytope and the actual reachable set can be made arbitrarily small. Let us remark that the polytopic over-approximation $\overline{\Omega}_i$ is not computed from the previous polytope of the sequence but from the support function of Ω_i. As a consequence, the proposed algorithm is not subject to the wrapping effect.

We now consider the problem of computing efficiently these polytopic over-approximations of the reachable set. We present an efficient algorithm that evaluates the support functions $\rho_{\Omega_0}, \ldots, \rho_{\Omega_{N-1}}$ in a given direction ℓ. It is based on the same observation as that made for Algorithm 2. Let us introduce the following auxiliary sequences $r_0, \ldots, r_{N-1} \in \mathbb{R}^n$ and $s_0, \ldots, s_{N-1} \in \mathbb{R}$:

$$\begin{array}{rclcrcl} r_0 & = & \ell, & & r_{i+1} & = & \Phi_\tau^\top r_i, \\ s_0 & = & 0, & & s_{i+1} & = & s_k + \rho_{W_\tau}(r_i). \end{array}$$

Equivalently, we have

$$r_i = (\Phi_\tau^\top)^i \ell \text{ and } s_i = \sum_{j=0}^{i-1} \rho_{W_\tau}\left((\Phi_\tau^\top)^j \ell\right).$$

Therefore,

$$\rho_{\Omega_i}(\ell) = \rho_{\Omega_0}(r_i) + s_i.$$

Algorithm 3 implements efficiently the evaluation of $\rho_{\Omega_0}(\ell), \ldots, \rho_{\Omega_{N-1}}(\ell)$. It performs, at each of the $N-1$ iterations, the product of a matrix with a vector and the evaluation of the support functions ρ_{Ω_0} and ρ_{W_τ} given by (5.8) and (5.7). The global time and space complexity of Algorithm 3 is therefore linear in the number of steps N.

Algorithm 3 Evaluation of $\rho_{\Omega_0}(\ell), \ldots, \rho_{\Omega_{N-1}}(\ell)$.

Require: The matrix Φ_τ given by (5.6), the support functions ρ_{Ω_0} and ρ_{W_τ} given by (5.8) and (5.7), the vector ℓ and an integer N.
Ensure: $\rho_i = \rho_{\Omega_i}(\ell)$ for i in $\{0, \ldots, N-1\}$

```
r_0 ← ℓ
s_0 ← 0
ρ_0 ← ρ_{Ω_0}(r_0)
for i from 0 to N − 2 do
    r_{i+1} ← Φ_τ^⊤ r_i
    s_{i+1} ← s_i + ρ_{W_τ}(r_i)
    ρ_{i+1} ← ρ_{Ω_0}(r_{i+1}) + s_{i+1}
end for
return {ρ_0, ..., ρ_{N−1}}
```

5.1.4 Numerical examples

A five-dimensional linear system: As a first benchmark consider the five-dimensional example taken from [Gir05]. Over-approximations of the reachable sets of this system have been computed using Algorithm 1 with a reduction step to keep the number of generators less than 100, Algorithms 2 and 3 where the facets of the approximating polytopes are chosen aligned with the axes. The approximation obtained by Algorithm 2 is always the most accurate because it exactly implements the recurrence relation (5.2). For short time horizons, the over-approximations computed by Algorithm 1 are more accurate than the ones computed by Algorithm 3. However, as we consider longer time horizons, the errors introduced at each step of Algorithm 1 start propagating through the computations

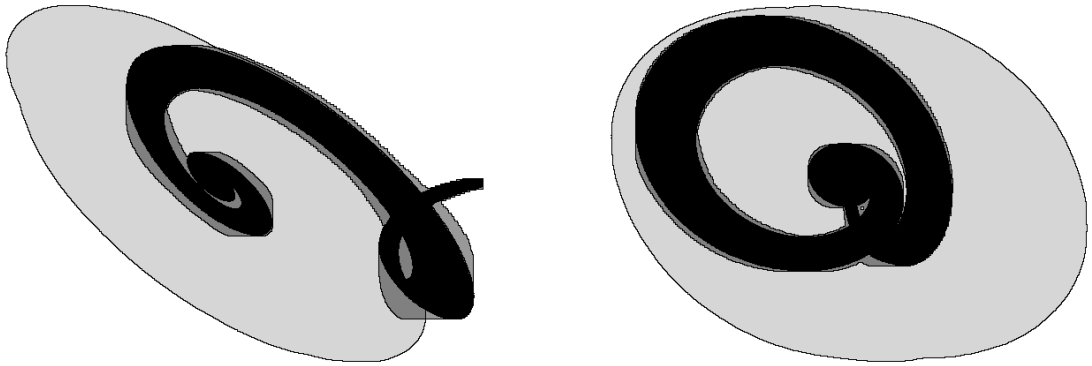

Figure 5.3: Reachable states of a five-dimensional linear system after 1000 iterations: projections on coordinates x_1 and x_2 (left), x_4 and x_5 (right). In light gray: set computed by Algorithm 1 with a reduction step to keep the number of generators smaller than 100. In dark gray: set computed by Algorithm 3 where the facets of the approximating polytopes are chosen aligned with the axes. In black: set computed by Algorithm 2.

and the wrapping effect becomes too significant to actually say anything interesting about the reachable states of the system. In comparison, the over-approximations obtained by Algorithm 3 are tight and remain accurate even for long time horizons.

Figure 5.3 shows the over-approximations of the reachable sets obtained by the three algorithms for a long time horizon ($N = 1000$). It is clear that Algorithms 2 and 3 have a much better precision than Algorithm 1, an obvious victim of the wrapping effect. Algorithm 1 needs 2.16s and 6.88MB to compute the over-approximations, for the same task Algorithms 2 and 3 need 0.07s and 1.47MB, 0.02s and 246KB, respectively.

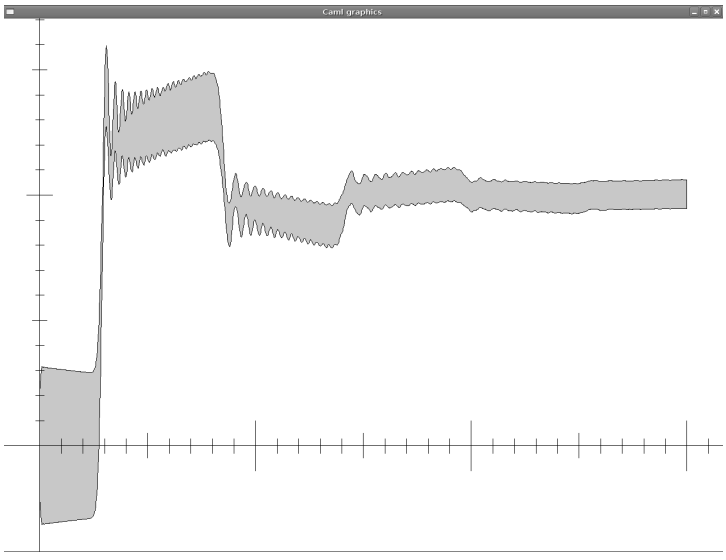

Figure 5.4: Reachable values by $u_{out}(t)$ against time t.

RLC model of a transmission line: We consider a second example consisting in a verification problem for a transmission line borrowed from [HK06]. The goal is to check that the transient behavior of a long transmission line is acceptable both in terms of overshoot and of response time. The dynamics of the system is given by a single-input single-output linear dynamical system with a state vector in dimension $n = 81$. Initially, the system is supposed to be in an ε-neighborhood (with $\varepsilon = 0.01$) of the set of steady states for an input value inside $[-0.2; 0.2]$. Then, at time $t = 0$, the input value is switched to a value in $[0.99; 1.01]$. Figure 5.4 shows the reachable values of the output voltage for a time horizon of 3ns, it was computed by Algorithm 3 in 0.10s using 0.234MB.

5.2 Reachability Analysis of Polynomial Systems

We now consider a disrete-time dynamical system of the following form:

$$x_{k+1} = f(x_k),\ k \in \mathbb{N},\ x_k \in \mathbb{R}^n,\ x_0 \in X_0 \tag{5.9}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a polynomial map and X_0 is a bounded polytope in $\mathbb{R}^n$.

We are concerned with bounded-time reachability analysis of system (5.9) which consists in computing the sequence $X_k \subseteq \mathbb{R}^n$ of reachable sets at time k of the system up to some time $K \in \mathbb{N}$. It should be noticed that even though the first element X_0 is a polytope, in general the other elements of the sequence are not. Actually, they are generally not even convex. In the following, we present an approach [BTDG12] for computing over-approximations of the sets X_k using bounded polytopes $\overline{X}_k$. Such a sequence can clearly be computed inductively by setting $\overline{X}_0 = X_0$ and by ensuring that for all $k = 0, \ldots, K-1$, $f(\overline{X}_k) \subseteq \overline{X}_{k+1}$.

Hence, we focus on the computation of a polytopic over-approximation $\overline{X}_{k+1}$ of the image of a bounded polytope $\overline{X}_k$ by the map f. We seek $\overline{X}_{k+1}$ under the form

$$\overline{X}_{k+1} = \{x \in \mathbb{R}^n |\ A_{k+1}x \leq b_{k+1}\}.$$

where the direction matrix $A_{k+1} \in \mathbb{R}^{m \times n}$, the position vector $b_{k+1} \in \mathbb{R}^m$ and the inequality above is to be understood component-wise. Let us assume for the moment that the direction matrix A_{k+1} is given. Then, the computation of the set $\overline{X}_{k+1}$ reduces to determining value b_{k+1}. Then, if for all $i = 1, \ldots, m$

$$-b_{k+1,i} \leq \min_{x \in \overline{X}_k} -A_{k+1,i} f(x) \tag{5.10}$$

it follows that $f(\overline{X}_k) \subseteq \overline{X}_{k+1}$.

Let us remark that the computation of the minimal values in equation (5.10) involves optimizing a generally non-convex multi-variable polynomial function on a bounded polytope. This is a difficult problem in general; however the computation of a lower bound for the minimal values is sufficient to obtain an over-approximation of $f(\overline{X}_k)$. Accurate lower bounds can be computed by solving semi-definite programs obtained through sum of squares or linear matrix inequalities relaxations [Par03, Las01]. However, since the number of optimization problems to be solved can be quite large (one problem by facet and by time step), one may be interested in obtaining less accurate but cheaper lower bounds. For that purpose, a technique, based on linear programming, is presented in the following section.

5.2.1 Optimization of polynomials using linear programming

In the following, we consider the problem of computing a guaranteed lower bound of the following optimization problem:

$$\begin{array}{ll} \text{minimize} & \ell \cdot g(y) \\ \text{over} & y \in [0,1]^n, \\ \text{subject to} & Ay \leq b. \end{array} \tag{5.11}$$

where $\ell \in \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^n$ is a polynomial map, $A \in \mathbb{R}^{n \times m}$ and $b \in \mathbb{R}^m$. It is clear that the optimization problem in (5.10) can be written under the form (5.11) using a linear change of variables.

Let $y_1, \ldots, y_n$ denote the components of $y \in [0,1]^n$ and $\delta_1, \ldots, \delta_n$ denote the degrees of g in $y_1, \ldots, y_n$. Let $\Delta = (\delta_1, \ldots, \delta_n)$; for $I = (i_1, \ldots, i_n) \in \mathbb{N}^n$, we write $I \leq \Delta$ if $i_j \leq \delta_j$ for all $j \in \{1, \ldots, n\}$. The main ingredient of our approach is the Bernstein expansion of polynomials. The polynomial map g in its Bernstein form is given by:

$$g(y) = \sum_{I \leq \Delta} h_I B_{\Delta,I}(y) \text{ where } h_I \in \mathbb{R}^n, \ \forall I \leq \Delta$$

and the Bernstein polynomials are defined for $I \leq \Delta$ as follows:

$$B_{\Delta,I}(y) = \beta_{\delta_1,i_1}(y_1) \ldots \beta_{\delta_n,i_n}(y_n)$$

with for $j = 1, \ldots n$, $i_j = 0, \ldots, \delta_j$: $\beta_{\delta_j,i_j}(y_j) = \left(\begin{array}{c} \delta_j \\ i_j \end{array} \right) y_j^{i_j} (1 - y_j)^{\delta_j - i_j}$.

For determining a linear programming relaxation of (5.11), the most useful properties of the Bernstein polynomials are the following:

Proposition 5.8 *The Bernstein polynomials satisfy the following properties:*

1. *For all* $y \in \mathbb{R}^n$, $\sum_{I \leq \Delta} B_{\Delta,I}(y) = 1$ *and* $\sum_{I \leq \Delta} \frac{I}{\Delta} B_{\Delta,I}(y) = y$.
2. *For all* $y \in [0,1]^n$, $0 \leq B_{\Delta,I}(y) \leq B_{\Delta,I}(\frac{I}{\Delta})$
   where $B_{\Delta,I}(\frac{I}{\Delta}) = \prod_{j=1}^{j=n} \left(\begin{array}{c} \delta_j \\ i_j \end{array} \right) \frac{i_j^{i_j} (\delta_j - i_j)^{\delta_j - i_j}}{\delta_j^{\delta_j}}$.

We can use the previous proposition to derive a linear programming relaxation of the problem (5.11):

Proposition 5.9 *Let* $\underline{p}^*$ *be the optimal value of the linear program:*

$$\begin{array}{lll} \textit{minimize} & \sum_{I \leq \Delta} (\ell \cdot h_I) z_I & \\ \textit{over} & z_I \in \mathbb{R}, & I \leq \Delta, \\ \textit{subject to} & 0 \leq z_I \leq B_{\Delta,I}(\frac{I}{\Delta}), & I \leq \Delta, \\ & \sum_{I \leq \Delta} z_I = 1, & \\ & \sum_{I \leq \Delta} (A \frac{I}{\Delta}) z_I \leq b. & \end{array}$$

Then, $\underline{p}^* \leq p^*$ *where* p^* *is the optimal value of problem (5.11).*

Thus, we can see that the computation of $\overline{X}_{k+1}$ can be done by computing guaranteed lower bounds on the optimal values of minimization problems using linear programming.

5.2.2 Choice of the direction matrices

Let the polytope $\overline{X}_k = \{x \in \mathbb{R}^n |\ A_k \cdot x \leq b_k\}$, in the next iteration, we want to compute a new direction matrix A_{k+1} that reflects as much as possible the changes of the shape of $\overline{X}_k$ under the polynomial f. For that purpose, we use a local linear approximation of the dynamics of the polynomial dynamical system (5.9) given by the first order Taylor expansion around an element x_k^* of the last computed polytope $\overline{X}_k$:

$$f(x) \ \approx L_k(x) = f(x_k^*) + J(x_k^*)(x - x_k^*)$$

where J is the Jacobian matrix of the function f. Let us denote $F_k = J(x_k^*)$ and $h_k = f(x_k^*) - J(x_k^*)x_k^*$, then in a neighborhood of x_k^* the nonlinear dynamics can be roughly approximated by $x_{k+1} = F_k x_k + h_k$. Assuming that F_k is invertible, this gives $x_k = F_k^{-1}x_{k+1} - F_k^{-1}h_k$. Transposing the constraints on x_k given by $\overline{X}_k$ to x_{k+1}, we obtain

$$A_k F_k^{-1} x_{k+1} \leq b_k + A_k F_k^{-1} h_k$$

Then, it appears that a reasonable template for $\overline{X}_{k+1}$ should be $A_{k+1} = A_k F_k^{-1}$. This new template A_{k+1} can then be used in next iteration for the computation of the polytope $\overline{X}_{k+1}$ using the method described in the previous section. Let us remark that this choice implies that our reachability algorithm is exact if f is an affine map.

Example: We consider a discrete time version of the FitzHugh-Nagumo model which is a polynomial dynamical system modelling the electrical activity of a neuron:

$$\begin{cases} x_1(k+1) &= x_1(k) + h\left((x_1(k) - \frac{x_1(k)^3}{3} - x_2(k) + I\right) \\ x_2(k+1) &= x_2(k) + h\,(0.08(x_1(k) + 0.7 - 0.8x_2(k))) \end{cases}$$

where the model parameter I is equal to $\frac{7}{8}$ and the time step $h = 0.05$. Figure 5.5 shows two reachable set evolutions. The figure on the left was computed using static direction matrices $A_k = A_0$, for all k whereas the figure on the right was computed using dynamic direction matrices as described above. The computation time is 1.16 seconds using static direction matrices and 1.22 seconds using dynamic direction matrices. We can see from the figure a significant precision improvement obtained by using dynamical templates, at little additional cost.

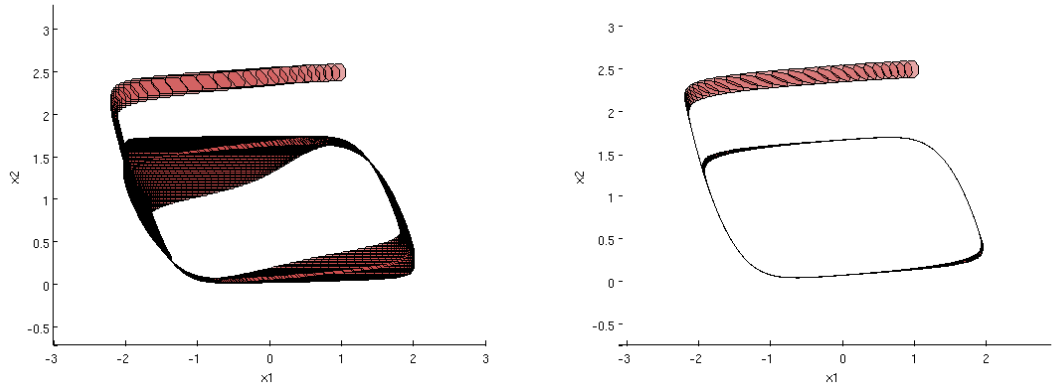

Figure 5.5: Reachability sets for the FitzHugh-Nagumo neuron model using static (left) and dynamic (right) direction matrices.

Discussion: The results presented in Section 5.1 were published in [Gir05, GLM06, LG10]. The time-discretization scheme was developed in collaboration with Colas Le Guernic during his Ph.D. at Verimag under the co-supervision of Oded Maler and myself; it is an improvement of an earlier scheme published in [Gir05]. The first implementation using zonotopes is taken from [Gir05], the improved algorithmic scheme and the approach based on support functions were developed in collaboration with Colas Le Guernic and Oded Maler and presented in [GLM06, LG10]. The zonotope based approach has been extended to handle linear systems with uncertain parameters [ASB07] and nonlinear systems using local linearizations [ASB08]. The approach using support functions has been adapted to compute viability kernels of linear systems in [MKM$^+$13]. Finally, let us remark that zonotopes and support functions have recently been used as abstract domains for static analysis of numerical programs [GGP09, SB13].

The extension of the algorithms of Section 5.1 for reachability analysis of hybrid automata has been described in [GL08] (see also [ASB10]) for the approximation based on zonotopes and in [LG09] for the approximation based on support functions. The main difficulty of the extension lies in the computation of the intersection of the continuous reachable sets with the invariants and with the guards enabling the discrete transitions. It is noticeable that the support function algorithm constitutes the core of the state of the art hybrid system verification platform SpaceEx [FLD$^+$11] developed in Verimag by a team led by Goran Frehse.

The approach for polynomial systems presented in Section 5.2 was developed within the ANR project VEDECY. The linear programming relaxations of polynomial optimization problems were developed by Mohamed Amin Ben Sassi during his Ph.D. under the co-supervision of Guillaume James and myself. The algorithm for reachability analysis was presented in [BTDG12] and developed in collaboration with Romain Testylier and Thao Dang from Verimag and improves the existing reachability algorithms based on the Bernstein form [DS09]. Other applications of linear programming relaxations of polynomial optimization were considered by Mohamed Amin Ben Sassi during his Ph.D. such as the computation of polytopic invariants [BG12a] or synthesis of robust controllers [BG12b] for continuous-time polynomial dynamical systems.

Reachability analysis of general nonlinear systems can be performed using the hybridization principle, which consists in approximating a nonlinear dynamics by a piecewise linear dynamics where additional bounded disturbances are added to account for the approximation error. Then, the reachable set of the resulting hybrid automaton, computable with approaches mentioned above, provides an over-approximation of the reachable set of the original non-linear system. In collaboration with Eugène Asarin from LIAFA and Thao Dang, we have shown that the approximation error of the reachable set can be made arbitrarily accurate on bounded time intervals and even on unbounded time intervals if an attractor exists [ADG03, ADG07]. The hybridization principle has also been applied to controller synthesis for nonlinear systems in collaboration with Samuel Martin while he was a student at ENSIMAG [GM12].

Part III

Multi-Agent Dynamical Systems

Chapter 6

Consensus and Opinion Dynamics

Résumé : *Les systèmes dynamiques multi-agents consistent en un ensemble d'agents inter-agissant localement par des règles simples afin de réaliser un comportement collectif cohérent. Ils trouvent des applications dans de nombreux domaines tels que la robotique, la conception des réseaux d'énergie ou en sciences sociales. Le problème du consensus constitue un point central de la théorie des systèmes multi-agents. On parle de consensus lorsque l'ensemble des agents converge vers un même état. Le défi principal consiste à trouver des conditions sur la structure des interactions locales entre agents qui garantissent l'obtention d'un consensus. Nous présentons un résultat [MG13] pour le consensus en temps continu qui propose des conditions suffisantes (connexité persistante et divergence lente des poids d'interaction réciproques) pour le consensus qui sont plus générales que celles disponibles aujourd'hui dans la littérature. De plus, notre approche permet d'établir une estimation de la vitesse de convergence vers le consensus. Nous présentons ensuite un modèle de dynamique d'opinion permettant de reproduire la formation de communautés au sein d'un réseau d'agents [MG11]. Dans le modèle considéré, les agents cherchent à atteindre un consensus avec une contrainte sur la vitesse de convergence vers celui-ci. Lorsque la vitesse prescrite ne peut être garantie, les interactions entre agents ne s'accordant pas assez vite sont supprimées. Il résulte de ce modèle la formation de sous-groupes d'agents (les communautés) au sein desquels un consensus est atteint. Nous établissons une caractérisation de ces communautés en terme de propriétés algébriques du graphe d'interactions et décrivons une application de notre modèle à la détection de communautés dans des graphes. Ainsi, notre modèle peut être utilisé comme algorithme décentralisé pour la détection de communautés dans les réseaux.*

Multi-agent dynamical systems consists of agents interacting according to simple local rules in order to achieve some global coordinated behavior. These systems find numerous applications such as multi-vehicle control in robotics [RBA07], design of smart distributed energy networks [DCB13], modeling of opinion dynamics [HK02]... The capability of reaching an agreement in a distributed manner is a central problem in multi-agent systems; consensus algorithms serve to emulate this process of agreement: agents locally exchange information with their neighbors about their states (representing e.g. positions

and velocities, power production, or opinions depending on the considered application) and act in order to decrease the distance between these. A multi-agent system is said to reach a consensus when the states of all agents converge asymptotically toward a common value.

Sufficient conditions for convergence to a consensus are typically based on the topology of the network (or graph) describing the interactions between neighbors and on the strength of these interactions. Consensus algorithms have attracted a lot of attention in the past decade. Notable convergence results include [JLM03, Mor05, RB05, BHOT05, HT12] for the discrete time and [OSM04, Mor04, RB05, HT12, CZZ11a, CZZ11b] for the continuous time consensus algorithm. We can classify these results depending on whether or not they require some notion of reciprocity in the interaction. In the following, we assume some kind of reciprocity which allows us to consider weaker assumptions on the connectivity of the interaction graph. In the first part of this chapter, we present a set of sufficient conditions for achieving consensus in continuous-time. These are the weakest conditions available in the literature for continuous-time consensus with some reciprocity in the interactions. In particular, they extend the very recent result from [HT12]. Moreover, our result provides an explicit bound on the convergence rate to consensus which is missing in [HT12]. Numerical examples are shown to illustrate the tightness of our conditions.

In the second part of the chapter, we present a model of opinion dynamics with decaying confidence. It is a discrete-time multi-agent system where the state of each agent represents its opinion. At each time step, the agent receives the opinions of its neighbors and then updates its opinion by taking a weighted average of its opinion and the opinions of its neighbors that are within some confidence range of its own. The confidence ranges are getting smaller at each time step: an agent gives repetitively confidence only to the neighbors that approach sufficiently fast its own opinion. This can be seen as a model for a negotiation process where an agent expects that its neighbors move significantly towards its opinion at each negotiation round in order to keep negotiating. Our model can be seen as an extension of the opinion dynamics with bounded confidence studied in [HK02, BHT09]. In our model, global consensus may not be achieved and the agents may only reach local agreement. We call communities the subsets of agents reaching a consensus. We provide an algebraic characterization of these communities and we show that our model provides a naturally distributed solution to the community detection problem in graphs.

6.1 Sufficient Conditions for Consensus

The system we study consists of n *agents* interacting with each other according to a continuous-time consensus protocol. Agents are labeled from 1 to n and $V = \{1, \dots, n\}$ denotes the label set of the agents. Agents adjust their *states* $x_i(t) \in \mathbb{R}$ for $i \in V$ according to the following differential equation

$$\dot{x}_i(t) = \sum_{j=1}^{n} a_{ij}(t)(x_j(t) - x_i(t)),\ i \in V \tag{6.1}$$

where for all $i, j \in V$, the *interaction weight* a_{ij} represents the strength of the influence of agent j on agent i and is a non-negative measurable function of time, summable on bounded intervals of $\mathbb{R}^+$. We call a solution to equation (6.1), a *trajectory* of the system.

We say that a trajectory *reaches a consensus* when $\lim_{t\to+\infty} x_i(t)$ exist and are the same, for all $i \in V$. The common limit is called the *consensus value*. We define the *group diameter* as

$$\Delta_V(t) = \max_{i\in V} x_i(t) - \min_{j\in V} x_j(t).$$

It can be easily shown that $\max_{i\in V} x_i(t)$ is non-increasing and that $\min_{j\in V} x_j(t)$ is non-decreasing. Then, it is clear that the group diameter is non-increasing and that the trajectory reaches a consensus if and only if $\lim_{t\to+\infty} \Delta_V(t) = 0$.

6.1.1 Persistent connectivity and slow divergence of reciprocal interaction weights

In this section, we present a set of sufficient conditions for achieving consensus [MG13]. Our convergence result involves assumptions on the interaction weights. Let S be some non-empty proper subset of V, we define the ratio between reciprocal interaction weights from S to $V \setminus S$:

$$r_S(t) = \begin{cases} \frac{\sum_{i\in S, j\notin S} a_{ij}(t)}{\sum_{i\in S, j\notin S} a_{ji}(t)} & \text{if the denominator is positive,} \\ 1 & \text{if numerator and denominator are equal to zero,} \\ +\infty & \text{if the denominator is zero and the numerator is positive.} \end{cases}$$

Then, we define the maximal ratio between reciprocal interaction weights as follows:

$$r(t) = \max_{S\neq\emptyset, S\subsetneq V} r_S(t)$$

For manipulation purposes, we shall use the maximal value of r over all past times. Let

$$\mathbf{r}(t) = \sup_{s\in[0,t]} r(s).$$

As defined, $\mathbf{r}$ is a non-decreasing function of time, and $\mathbf{r}(t)$ is always greater than 1. A direct consequence of this definition is the following statement:

$$\forall S \neq \emptyset, S \subsetneq V,\ \forall s \in [0,t],\ \frac{1}{\mathbf{r}(t)} \sum_{i\in S, j\notin S} a_{ij}(s) \leq \sum_{i\in S, j\notin S} a_{ji}(s) \leq \mathbf{r}(t) \sum_{i\in S, j\notin S} a_{ij}(s).$$

Thus, whenever a subgroup S of agents influences the rest of the group via interaction weights of sum $a(s) > 0$ at time $s \leq t$, we know that S is influenced back via interaction weights of sum no less than $\frac{a(s)}{\mathbf{r}(t)}$.

In our convergence result, we shall make two assumptions on interaction weights. The first one is concerned with the topology of the interactions and involves the notion of strong connectivity. A graph, with V as set of vertices, is said to be strongly connected when there is a directed path going from node i to node j for all distinct nodes $i, j \in V$.

Assumption 6.1 (Persistent Connectivity) *The graph (V, E) is strongly connected where*

$$E = \left\{ (j,i) \in V \times V \mid \int_0^{+\infty} a_{ij}(s)ds = +\infty \right\}.$$

This assumption allows us to define a sequence of time instants $(t_p)_{p\in\mathbb{N}}$ which implicitly defines a rescaling of time according to the speed of growth of $\int_0^t a_{ij}(s)ds$ for $(j,i)\in E$. Let $t_0 = 0$ and, for $p \in \mathbb{N}$, let us define t_{p+1} as the last element of the intermediate finite sequence $(t_p^0, t_p^1, \ldots, t_p^{\lfloor n/2\rfloor})$ where $\lfloor\cdot\rfloor$ is the floor function, $t_p^0 = t_p$ and for $q \in \{0,\ldots,\lfloor n/2\rfloor\}$, t_p^{q+1} is the smallest time $t \geq t_p^q$ such that

$$\min_{S\subsetneq V, S\neq\emptyset}\left(\sum_{i\in S}\sum_{j\in V\setminus S}\int_{t_p^q}^{t} a_{ij}(s)ds\right) = 1.$$

Such a t always exists because (V,E) is strongly connected and therefore for all non empty set $S \subsetneq V$, there exists $i \in S$ and $j \in V\setminus S$ such that $(j,i)\in E$. Essentially, the sequence $(t_p^0, t_p^1, \ldots, t_p^{\lfloor n/2\rfloor})$ defines time intervals $[t_p^q, t_p^{q+1}]$ over which the cumulated influence on any subgroup of agents from the rest of the agents is no less than 1. Let us remark that since we assume that the interaction weights a_{ij} are summable on bounded intervals of $\mathbb{R}^+$, it follows that the sequence $(t_p)_{p\in\mathbb{N}}$ goes to infinity as p goes to $+\infty$. We can now state our main result that quantifies the contraction of the group diameter between time t_p and t_{p+1}:

Proposition 6.1 (Group diameter contraction rate) *If Assumption 6.1 (persistent connectivity) holds, then for all $p \in \mathbb{N}$,*

$$\Delta_V(t_{p+1}) \leq \left(1 - \frac{\mathbf{r}(t_{p+1})^{-\lfloor n/2\rfloor}}{(8n^2)^{\lfloor n/2\rfloor}}\right)\Delta_V(t_p).$$

The main idea of the rather technical proof of this proposition is to show that the interactions over intervals $[t_p^q, t_p^{q+1}]$ induce a chain of movements of the agents toward the center of the group. These movements propagate toward agents having either smallest or largest states in less than $\lfloor n/2\rfloor$ such intervals and result in a contraction of the group diameter between t_p and t_{p+1}.

It is clear from the previous proposition that the sequence $(\mathbf{r}(t_p))_{p\in\mathbb{N}}$ plays a central role in the fact that the consensus is reached or not. This is where the second assumption regarding the interaction weights comes into play.

Assumption 6.2 (Slow divergence of reciprocal interaction weights) *For all $t \geq 0$, $\mathbf{r}(t)$ is finite and the infinite sum $\sum_{p\in\mathbb{N}} \mathbf{r}(t_p)^{-\lfloor n/2\rfloor} = +\infty$.*

The assumption requires $\mathbf{r}(t)$ not to grow too fast. For instance, $\mathbf{r}(t_p) = O(p^{2/n})$ (which includes the case where $\mathbf{r}$ is bounded) satisfies Assumption 6.2, whereas $\mathbf{r}(t_p) = p^{4/n}$ does not. Hence, the assumption enables the divergence of reciprocal interaction weights provided this divergence is slow. Let us remark that the larger the number of agents, the slower the divergence can be. We can now state the main result of the paper.

Theorem 6.1 *If Assumptions 6.1 (persistent connectivity) and 6.2 (slow divergence of reciprocal interaction weights) hold, then the trajectory of system (6.1) reaches a consensus.*

The previous result provides the most general conditions available in the literature for continuous-time consensus with some assumption of reciprocity. In particular, it extends the very recent result [HT12] where Assumption 6.2 is essentially replaced by the assumption that $\mathbf{r}(t)$ is uniformly bounded by some constant $K > 0$. Moreover, thanks to different proof techniques, we are able to provide an estimation of the convergence rate by Proposition 6.1 which is missing in [HT12].

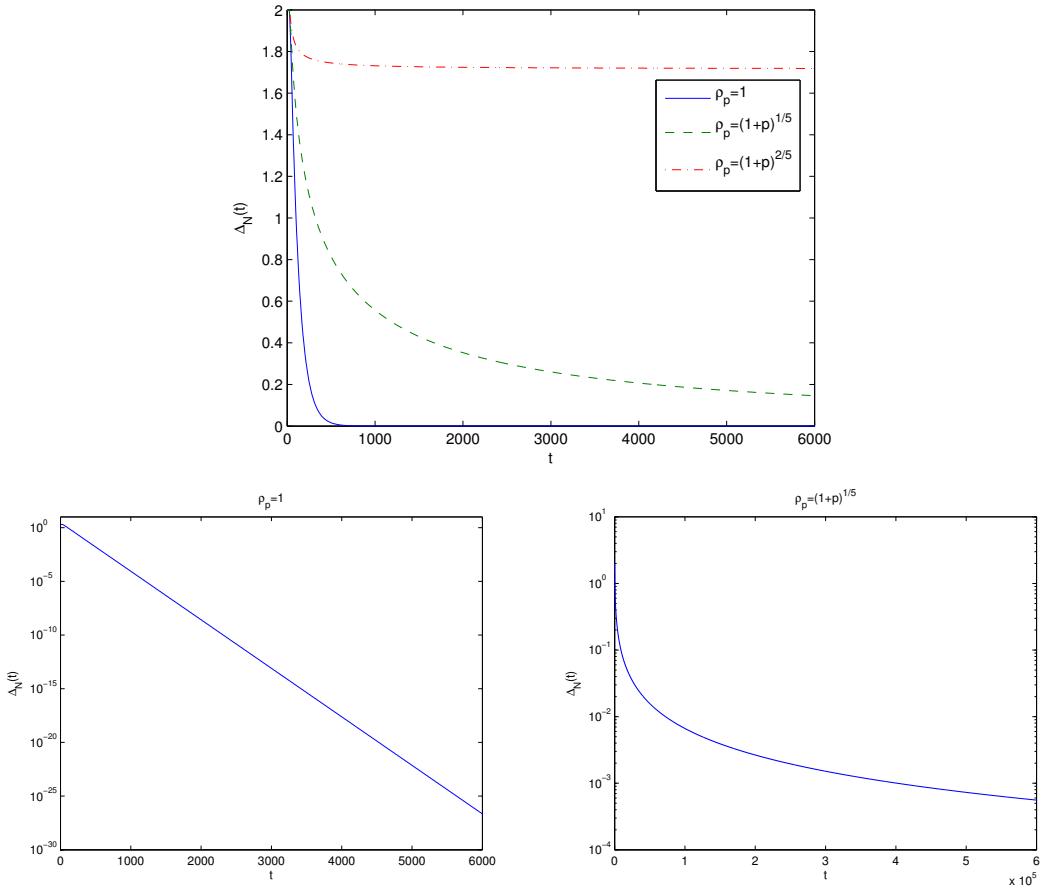

Figure 6.1: Top: evolution of the diameter $\Delta_V(t)$ for the system with 11 agents for the sequences given by $\rho_p = 1$ (plain line), $\rho_p = (1+p)^{\frac{1}{5}}$ (dashed line), $\rho_p = (1+p)^{\frac{2}{5}}$ (dash and dots). Bottom: evolution of the diameter $\Delta_V(t)$ in logarithmic scale for the sequences given by $\rho_p = 1$ (left), $\rho_p = (1+p)^{\frac{1}{5}}$ (right).

Numerical example: We realized a numerical study to evaluate the tightness of Assumption 6.2. Let us consider a system with $n = 2m+1$ agents with $m \geq 2$ whose dynamics is defined as follows:

- For $t \in [(m+1)p + i, (m+1)p + i + 1)$ with $p \in \mathbb{N}$, $i \in \{0, \ldots, m-2\}$,

$$\begin{cases} \dot{x}_{i+1}(t) &= x_{i+2}(t) - x_{i+1}(t) \\ \dot{x}_{i+2}(t) &= \rho_p(x_{i+1}(t) - x_{i+2}(t)) \\ \dot{x}_{n-1-i}(t) &= \rho_p(x_{n-i}(t) - x_{n-1-i}(t)) \\ \dot{x}_{n-i}(t) &= x_{n-1-i}(t) - x_{n-i}(t) \\ \dot{x}_j(t) &= 0 \qquad \text{if } j \notin \{i+1, i+2, n-1-i, n-i\} \end{cases}$$

- For $t \in [(m+1)p + m - 1, (m+1)p + m)$ with $p \in \mathbb{N}$,

$$\begin{cases} \dot{x}_m(t) &= x_{m+1}(t) - x_m(t) \\ \dot{x}_{m+1}(t) &= \rho_p(x_m(t) - x_{m+1}(t)) \\ \dot{x}_j(t) &= 0 \qquad\qquad \text{if } j \notin \{m, m+1\} \end{cases}$$

- For $t \in [(m+1)p + m, (m+1)(p+1))$ with $p \in \mathbb{N}$,

$$\begin{cases} \dot{x}_{m+1}(t) &= \rho_p(x_{m+2}(t) - x_{m+1}(t)) \\ \dot{x}_{m+2}(t) &= (x_{m+1}(t) - x_{m+2}(t)) \\ \dot{x}_j(t) &= 0 \qquad\qquad \text{if } j \notin \{m+1, m+2\} \end{cases}$$

This system satisfies the persistent connectivity Assumption 6.1. The sequence t_p grows linearly in p and we can also show that we have $\mathbf{r}(t_{p+1}) = \rho_p$, for $p \in \mathbb{N}$. Then, Assumption 6.2 holds if and only if $\sum_{p\in\mathbb{N}} {\rho_p}^{-m} = +\infty$.

In the following, we report the results of our numerical simulations of the system with 11 agents (i.e. $m = 5$). We simulated the system for three different sequences $(\rho_p)_{p\in\mathbb{N}}$: $\rho_p = 1$, $\rho_p = (1+p)^{\frac{1}{5}}$, $\rho_p = (1+p)^{\frac{2}{5}}$. It should be noted that Assumption 6.2 holds for the first two sequences but not for the third one. The results of the simulations are shown in Figure 6.1 where we represented the evolution of the diameter $\Delta_V(t)$ over time. The simulations are consistent with the theory showing that the consensus is reached for the first two sequences (the diameter goes to zero). Also, for the third sequence, we can observe that the consensus is not reached. For the first two sequences, we also represented the evolution of the diameter $\Delta_V(t)$ in a logarithmic scale in order to estimate the convergence rate. It appears clearly that for the first sequence the convergence rate is exponential. For the second sequence, the convexity of the curve indicates that the convergence rate is sub-exponential. These are consistent with the estimates given by Proposition 6.1. This example makes us think that our conditions for consensus are actually quite tight.

6.2 Opinion Dynamics with Decaying Confidence: Consensus in Communities

In this section we present a model of opinion dynamics with decaying confidence and its application to community detection in graphs [MG11].

6.2.1 Model description

We consider a set of n *agents*, $V = \{1, \ldots, n\}$. A relation $E \subseteq V \times V$ models the *interactions* between the agents. We assume that the relation is symmetric and anti-reflexive. V is the set of vertices and E is the set of edges of an undirected graph $G = (V, E)$, describing the network of agents. Each agent $i \in V$ has an *opinion* modeled by a real number $x_i(t) \in \mathbb{R}$. Initially, agent i has an opinion $x_i(0) = x_i^0$ independent from the opinions of the other agents. Then, at every time step, the agents update their opinion by taking a weighted average of its opinion and opinions of other agents:

$$x_i(t+1) = \sum_{j=1}^{n} p_{ij}(t) x_j(t) \tag{6.2}$$

with the coefficients $p_{ij}(t)$ satisfying

$$\forall i, j \in V,\ (p_{ij}(t) \neq 0 \iff j \in \{i\} \cup N_i(t)) \tag{6.3}$$

where $N_i(t)$ denotes the *confidence neighborhood* of agent i at time t:

$$N_i(t) = \left\{ j \in V |\ ((i,j) \in E) \wedge \left(|x_i(t) - x_j(t)| \leq R\rho^t \right) \right\} \tag{6.4}$$

with $R > 0$ and $\rho \in (0,1)$ are model parameters. We make the following assumption:

Assumption 6.3 *For $t \in \mathbb{N}$, the coefficients $p_{ij}(t)$ satisfy*

1. *$p_{ij}(t) \in [0,1]$, for all $i, j \in V$.*
2. *$\sum_{j=1}^{n} p_{ij}(t) = 1$, for all $i \in V$.*

This model can be interpreted in terms of opinion dynamics. At each time step t, agent $i \in V$ receives the opinions of its neighbors in the graph G. If the opinion of i differs from the opinion of its neighbor j more than the threshold $R\rho^t$, then i does not give confidence to j and does not take into account the opinion of j when updating its own opinion. The parameter ρ characterizes the confidence decay of the agents. Agent i gives repetitively confidence only to neighbors whose opinion converges sufficiently fast to its own opinion. This model can be interpreted in terms of negotiations where agent i requires that, at each negotiation round, the opinion of agent j moves significantly towards its own opinion in order to keep negotiating with j.

Remark 6.1 *We assume that $\rho \in (0,1)$, however, let us remark that for $\rho = 1$ (there is no confidence decay), with a complete graph G (every agent talks with all the other agents) and, for all $i \in V$, equal values of non-zero coefficients $p_{ij}(t)$, our model would coincide with Krause model of opinion dynamics with bounded confidence studied in [HK02, BHT09].*

Our first result states that the opinion of each agent converges to some limit value:

Proposition 6.2 *Under Assumption 6.3, for all $i \in V$, the sequence $(x_i(t))_{t \in \mathbb{N}}$ is convergent. We denote x_i^* its limit. Furthermore, we have for all $t \in \mathbb{N}$,*

$$|x_i(t) - x_i^*| \leq \frac{R}{1-\rho}\rho^t.$$

Generally, the opinions of all the agents do not converge to a common value. Indeed, agents may only succeed in agreeing locally organizing themselves in communities that we formally define as follows:

Definition 6.1 *Let $i, j \in V$, we say that agents i and j asymptotically agree, denoted $i \sim^* j$, if and only if $x_i^* = x_j^*$. $\sim^*$ is an equivalence relation over V, a community $C \subseteq V$ is an element of the quotient set $\mathcal{C} = V/\sim^*$.*

Let us remark that the community structure is dependent on the initial distribution of opinions. In the following, we shall provide an algebraic characterization of these communities.

6.2.2 Algebraic characterization of communities

Let us define the set of interactions at time t, $E(t) \subseteq V \times V$ as

$$E(t) = \left\{(i,j) \in E |\ |x_i(t) - x_j(t)| \leq R\rho^t\right\}.$$

Let us remark that $(i,j) \in E(t)$ if and only if $j \in N_i(t)$. The interaction graph at time t is then $G(t) = (V, E(t))$. Let us also define the *graph of communities* $G_{\mathcal{C}} = (V, E_{\mathcal{C}})$ where:

$$E_{\mathcal{C}} = \{(i,j) \in E |\ i \sim^* j\}.$$

We define the vectors of opinions and of initial opinions: $x(t) = (x_1(t), \ldots, x_n(t))^\top$ and $x^0 = (x_1^0, \ldots, x_n^0)^\top$. The dynamics of the vector of opinions is then given by

$$x(t+1) = P(t)x(t)$$

where $P(t)$ is the row stochastic matrix with entries $p_{ij}(t)$. For a set of agents $I \subseteq V$, with $I = \{v_1, \ldots, v_k\}$, $P_I(t)$ is the matrix with entries $p_{v_i v_j}(t)$. Let us remark that $P_I(t)$ is row stochastic if and only if $I \subseteq V$ is a subset of agents such that no agent in I is connected to an agent in $V \setminus I$ in the graph $G(t)$. We make the following assumption on the matrices $P(t)$:

Assumption 6.4 *The sequence of matrices $P(t)$ satisfy the following conditions:*

1. *For all $t \in \mathbb{N}$, $P(t)$ is invertible.*
2. *For all $t \in \mathbb{N}$, $P(t) = P(G(t))$.*

Let us remark that the first assumption can be enforced, for instance, by choosing $p_{ii}(t) > 1/2$ for all $i \in V$, for all $t \in \mathbb{N}$, in that case $P(t)$ is a strictly diagonally dominant matrix and therefore it is invertible. The second assumption states that $P(t)$ only depends on the graph $G(t)$, in particular this implies that since the set of subgraphs of G is finite, $P(t)$ can only take a finite number of values. This remark is fundamental for the proof of the result presented in this section.

We now make a last assumption. From Proposition 6.2, we know that the opinion of each agent converges to its limit value no slower than $O(\rho^t)$. This is an upper bound, the convergence to the limit value is actually often slightly faster. Let $X^0 \subseteq \mathbb{R}^n$ be the subset of vectors of initial opinions such that if $x^0 \in X^0$ then there exists $\underline{\rho} < \rho$ and $M \geq 0$ such that for all $i \in V$, for all $t \in \mathbb{N}$,

$$|x_i(t) - x_i^*| \leq M\underline{\rho}^t.$$

Let us remark that numerical experiments show that in practice $x^0 \in X^0$. This observation motivates the following assumption:

Assumption 6.5 *The vector of initial opinions x^0 is an element of X^0.*

It should be noted that unlike Assumptions 6.3, 6.4, it is generally not possible to check a priori whether Assumption 6.5 holds.

Let us consider a community $C \in \mathcal{C}$. Then it is clear that, in the graph $G_{\mathcal{C}}$, no agent in C is connected to an agent in $V \setminus C$. Therefore, it follows that $P_C(G_{\mathcal{C}})$ is a row stochastic

matrix. Then, let $\lambda_1(P_C(G_\mathcal{C})), \ldots, \lambda_{|C|}(P_C(G_\mathcal{C}))$ denote the eigenvalues of $P_C(G_\mathcal{C})$ with $\lambda_1(P_C(G_\mathcal{C})) = 1$ and

$$|\lambda_1(P_C(G_\mathcal{C}))| \geq |\lambda_2(P_C(G_\mathcal{C}))| \geq \cdots \geq |\lambda_{|C|}(P_C(G_\mathcal{C}))|.$$

The following theorem gives a characterization of the communities in terms of the eigenvalues $\lambda_2(P_C(G_\mathcal{C}))$ for $C \in \mathcal{C}$.

Theorem 6.2 *Under Assumptions 6.3 and 6.4, for almost all vectors of initial opinions $x^0 \in X^0$, for all communities $C \in \mathcal{C}$, such that $|C| \geq 2$, $|\lambda_2(P_C(G_\mathcal{C}))| < \rho$.*

A stronger version of Theorem 6.2 would state that the algebraic characterization of communities holds for almost all $x^0 \in \mathbb{R}^n$. To prove this result, we need to establish that $\mathbb{R}^n \setminus X^0$ is a set of zero measure, at least for generic values of ρ. We were not able to prove this result so far; however, experimental results tend to show that it holds in practice. In the following, we use Theorem 6.2 to address the problem of community detection in graphs.

6.2.3 Community detection in graphs via opinion dynamics

In the usual sense, communities in a graph are groups of vertices such that the concentration of edges inside communities is high with respect to the concentration of edges between communities. The community detection problem has attracted a lot of attention in the recent years. Some formalizations of the community detection problem have been proposed in terms of optimization of quality functions such as modularity ([NG04]).

The modularity of a partition measures how well the partition reflects the community structure of a graph. More precisely, let $G = (V, E)$ be an undirected graph, let $\mathcal{P}$ be a partition of V. Essentially, the modularity $\mathsf{Q}(\mathcal{P})$ of the partition $\mathcal{P}$ is the proportion of edges within the classes of the partition minus the expected proportion of such edges (see [NG04] for more details). The higher the modularity, the better the partition reflects the community structure of the graph. Thus, it is reasonable to formulate the community detection problem as modularity maximization. However, [BDG$^+$08] have shown that this optimization problem is NP-complete. Therefore, approaches for community detection rely mostly on heuristic methods. [New06] proposed a modularity optimization algorithm based on spectral relaxations. [BGLL08] presented a hierarchical combinatorial approach for modularity optimization. This algorithm which can be used for very large networks, is currently the one that obtains the partitions with highest modularity.

In the following section, we propose an alternative formulation of the community detection problem using a measure of connectivity of graphs given by the eigenvalues of their *normalized Laplacian matrix*.

Problem formulation

Let $G = (V, E)$ be an undirected graph with $V = \{1, \ldots, n\}$, with $n \geq 2$. For a vertex $i \in V$, the degree $d_i(G)$ of i is the number of neighbors of i in G. The normalized Laplacian of the graph G is the matrix $L(G)$ given by

$$L_{ij}(G) = \begin{cases} 1 & \text{if } i = j \text{ and } d_i(G) \neq 0, \\ \frac{-1}{\sqrt{d_i(G) d_j(G)}} & \text{if } (i,j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Let us review some of the properties of the normalized Laplacian matrix. $\mu_1(L(G)) = 0$ is always an eigenvalue of $L(G)$, it is simple if and only if G is connected. All other eigenvalues are real and belong to the interval $[0, 2]$. The second smallest eigenvalue of the normalized Laplacian matrix is denoted $\mu_2(L(G))$. It can serve as an algebraic measure of the connectivity: $\mu_2(L(G)) = 0$ if the graph G has two distinct connected components, $\mu_2(L(G)) = n/(n-1)$ if the graph is the complete graph (for all $i, j \in V$, $i \neq j$, $(i, j) \in E$), in the other cases $\mu_2(L(G)) \in (0, 1]$.

Let $\mathcal{P}$ be a partition of the set of vertices V. For all $I \in \mathcal{P}$, with $|I| \geq 2$, let $G_I = (I, E_I)$ be the subgraph of G consisting of the set of vertices I and of the set of edges of G between elements of I (i.e. $E_I = E \cap (I \times I)$). Let $L(G_I)$ denote the normalized Laplacian matrix of the graph G_I. Let us define the following measure associated to the partition $\mathcal{P}$

$$\underline{\mu_2}(\mathcal{P}) = \min_{I \in \mathcal{P}, |I| \geq 2} \mu_2(L(G_I)).$$

Essentially, $\underline{\mu_2}(\mathcal{P})$ measures the connectivity of the less connected component of $G_{\mathcal{P}}$. We now propose a formulation of the community detection problem:

Problem 6.1 *Given a graph $G = (V, E)$ and a real number $\delta \in (0, 1]$, find a partition $\mathcal{P}$ of V such that for all $I \in \mathcal{P}$, such that $|I| \geq 2$, $\mu_2(L(G_I)) > \delta$ (i.e. $\underline{\mu_2}(\mathcal{P}) > \delta$).*

If $\mu_2(L(G)) > \delta$, it is sufficient to choose the trivial partition $\mathcal{P} = \{V\}$. If $\delta \geq \mu_2(L(G))$, then we want to find groups of vertices that are more densely connected together than the global graph. This coincides with the notion of community. The larger δ the more densely connected the communities. This makes it possible to search for communities at different scales of the graph.

Let us remark that Problem 6.1 generally has several solutions. Actually, the trivial partition $\mathcal{P} = \{\{1\}, \ldots, \{n\}\}$ is always a solution. In the following, we show how non-trivial solutions to Problem 6.1 can be obtained using a model of opinion dynamics with decaying confidence. We evaluate the modularity of the partitions we obtain and compare our results to those obtained using modularity optimization algorithms presented in [New06, BGLL08].

Solution based on opinion dynamics

Let $\alpha \in (0, 1/2)$, we consider the model of opinion dynamics with decaying confidence defined by:

$$x_i(t+1) = \begin{cases} x_i(t) + \frac{\alpha \sum_{j \in N_i(t)} (x_j(t) - x_i(t))}{|N_i(t)|} & \text{if } N_i(t) \neq \emptyset \\ x_i(t) & \text{if } N_i(t) = \emptyset \end{cases} \tag{6.5}$$

where $N_i(t)$ is given by equation (6.4). It is straightforward to check that this model is a particular case of the model given by equations (6.2) and (6.3) and that Assumptions 6.3 and 6.4 hold. The following lemma relates the eigenvalues of $P_I(G_{\mathcal{P}})$ to that of $L(G_I)$.

Lemma 6.1 *Let $\mathcal{P}$ be a partition of V, $I \in \mathcal{P}$ such that $|I| \geq 2$. Then, λ is an eigenvalue of $P_I(G_{\mathcal{P}})$ if and only if $\mu = (1 - \lambda)/\alpha$ is an eigenvalue of $L(G_I)$.*

We now state the main result of the section which is a direct consequence of Theorem 6.2 and Lemma 6.1:

Proposition 6.3 *Let $\rho = 1 - \alpha\delta$, for almost all vectors of initial opinions $x^0 \in X^0$, the set of communities $\mathcal{C}$ obtained by the opinion dynamics model (6.5) is a solution to Problem 6.1.*

Example: books on American politics We propose to use our approach on an example consisting of a network of 105 books on politics initially compiled by V. Krebs (unpublished, see `www.orgnet.com`). In this network, each vertex represents a book on American politics bought from Amazon.com. An edge between two vertices means that these books are frequently purchased by the same buyer. The network is presented on the top left part of Figure 6.2 where the shape of the vertices represent the political alignment of the book (liberal, conservative, centrist).

δ	$\lvert\mathcal{C}\rvert$	$\underline{\mu_2}(\mathcal{C})$	$\mathsf{Q}(\mathcal{C})$	Occurrences
0.1	2	0.134	0.457	980
0.15	3	0.182	0.499	898
0.15	3	0.187	0.494	102
0.2	4	0.269	0.523	678
0.2	4	0.266	0.512	218

Table 6.1: Properties of the partitions of the books network (1000 different vectors of initial opinions for each value of parameter δ)

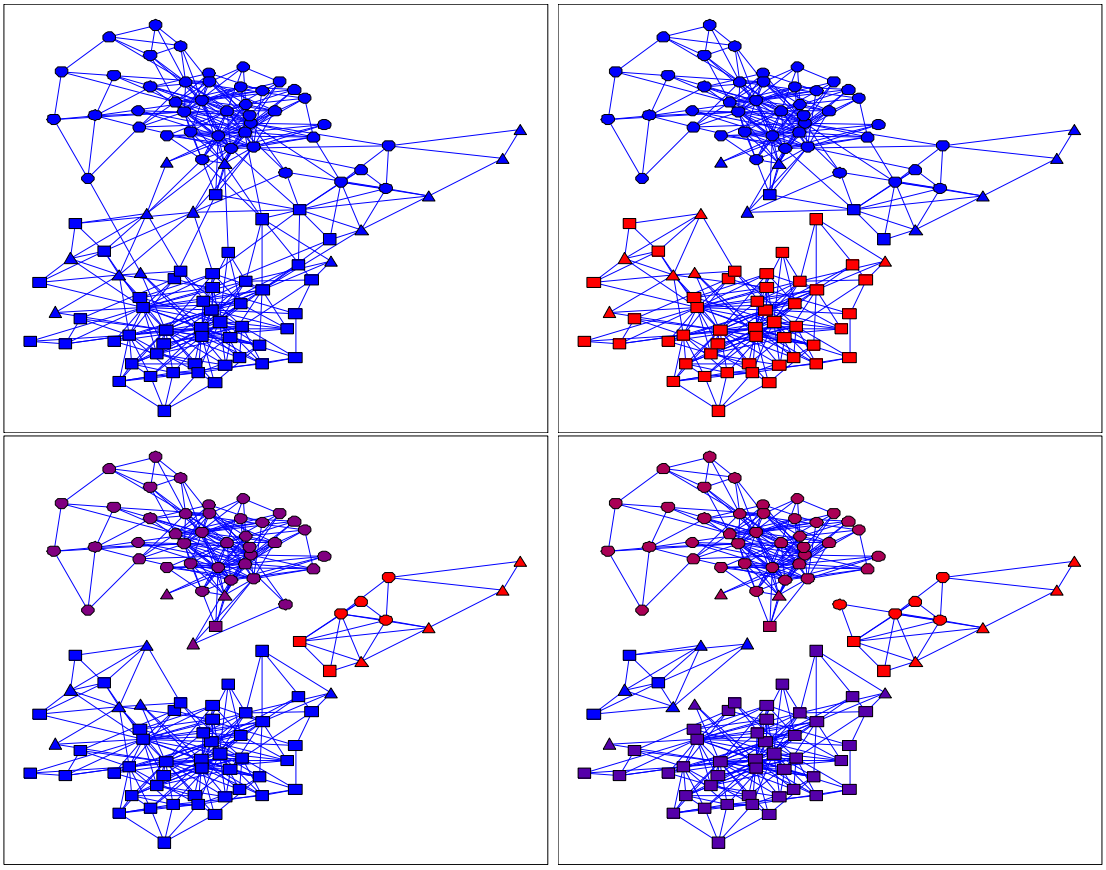

Figure 6.2: Graphs $G_{\mathcal{C}}$ for the most frequently obtained partition of the books network from top to bottom: initial graph, $\delta = 0.1$, $\delta = 0.15$, $\delta = 0.2$. Shapes represent political alignment of the books: circles are liberal, squares are conservative, triangles are centrist.

We used our opinion dynamics model (6.5) to uncover the community structure of this network. We chose 3 different values for δ. The parameters of the model are the same than in the previous example: $\alpha = 0.1$, $R = 1$ and $\rho = 1 - \alpha\delta$. For each different value of δ, the model was simulated for 1000 different vectors of initial opinions chosen randomly in $[0, 1]^{105}$. Simulations were performed as long as enabled by floating point arithmetics. The experimental results are reported in Table 6.1.

Let us remark that the computed partitions are solutions to the Problem 6.1. Also, for the same value of parameter δ, the modularity is very similar for all partitions. Actually, all the partitions obtained for the same value of δ are almost the same. The partition with maximal modularity is obtained for $\delta = 0.2$, it is a partition in 4 communities with modularity 0.523. As a comparison, algorithms in [New06] and [BGLL08] obtain partitions in 4 communities with modularity 0.526 and 0.527, respectively. As we can see, our partition has a modularity that is quite close from those obtained by these algorithms.

In Figure 6.2, we represented the graphs of communities $G_{\mathcal{C}}$ that are the most frequently obtained for the different values of δ. Let us remark that even though the information on the political alignment of the books is not used by the algorithm, our approach allows to uncover this information. Indeed, for $\delta = 0.1$, we obtain 2 communities that are essentially liberal and conservative. For $\delta = 0.2$, we then obtain 4 communities: liberal, conservative, centrist-liberal, centrist-conservative.

In this section, we have presented a model of opinion dynamics and we have shown that it can be used for community detection in graphs. The main advantage of our community detection algorithm with respect to existing ones is that it is distributed by nature. It can be ran concurrently by agents willing to determine which community they belong to, the agents do not need to know the global structure of the network.

Discussion: The results presented in the first part of the chapter have been developed in collaboration with Samuel Martin when he was a Ph.D. student at Laboratoire Jean Kuntzmann under the co-supervision of Guillaume James and myself. The proofs can be found in [MG13]. Within the thesis of Samuel Martin, we also considered the application of consensus protocols for controlling a group of vehicles. We have proposed a measure to estimate the robustness of a formation. This measure allows us to establish a bound on the velocity perturbation that can cause a disconnection of the interaction graph. The problem was first considered in the context of deterministic [MG10] interactions and extended to the stochastic case in collaboration with A. Jadbabaie and A. Fazeli from University of Pennsylvania [MFJG12].

The results on opinion dynamics and community detection were developed within the UJF MSTIC project CARESSE, in collaboration with Constantin Morarescu during his postdoctoral stay at Laboratoire Jean Kuntzmann. Proofs and additional cases studies can be found in the paper [MG11].

Chapter 7

Conclusion and Perspectives

This document presents the main contributions of my research work since 2004 within the areas of hybrid and multi-agent systems.

One characteristic in my research work has been the use of concepts and techniques that lie at the interface of several disciplines. The first part of this document deals with approximate simulation and bisimulation which are adaptations of classical tools in computer science. In order to compute symbolic abstractions that can serve for controller synthesis, these notions have been used in combination with Lyapunov techniques, which are widely developed in control theory. My work on reachability analysis mainly uses techniques from applied mathematics (numerical approximations, computational geometry, convex analysis...) but the whole approach has been inspired by the area of computer science called model checking. Finally, the whole field of multi-agent dynamical systems has been developed at the interface of graph and dynamical systems theory. I hope this document has succeeded in convincing the reader that there is a huge potential for the development of innovative approaches at the interface of applied mathematics, control theory and computer science.

Another characteristic of my work is the constant concern for algorithms and computation. Most of my theoretical contributions have been motivated by the development of computational approaches to analysis and control of hybrid systems. For instance, my work on approximately bisimilar abstractions has been concretized in the tool for controller synthesis CoSyMA; algorithms for reachability analysis using support functions constitutes the core of hybrid system verification platform SpaceEx developed at Verimag; my work on opinion dynamics has been motivated by the development of efficient distributed algorithms for community detection in large networks... I strongly believe that as we try to analyze and control dynamical systems of increasing complexity, computational techniques, possibly used in combination with analytical approaches, become an indispensable tool.

The work presented in this document draws inspirations for future development. Several problems that I plan to tackle are discussed below.

Advances in symbolic control

Input sequences as symbolic states: In Chapter 3, we have presented approaches for computing approximately bisimilar symbolic models for a class of incrementally stable switched systems. In these approaches, the computation of the symbolic models is based

on the use of discrete (uniform or multi-scale) lattices approximating the state-space. This discretization of the state-space clearly limits the application of our approach to dynamics of modest dimension. Intuitively, systems that are incrementally stable are those that have asymptotic forgetfulness, i.e. the effect of the initial condition on the current state of the system vanishes progressively. Hence, an alternative approach to the computation of symbolic models for incrementally stable switched systems consist in using finite mode sequences as symbolic states representing the concrete states that are reached by the switched system when the associated mode sequence is applied. This approach can be related to [TP06], where symbolic models for linear systems with bounded memory are computed by identifying symbolic states with bounded input sequences. In that case, the resulting symbolic models are related to the original system by an exact bisimulation relation. The fact that we do not explicitly discretize the state-space makes this approach potentially more suitable than those presented in this document for higher dimensional systems. Actually, it even opens the way to the computation of symbolic models for infinite dimensional systems modeled by partial differential equations. For that purpose, it will first be necessary to define and characterize the notion of incremental stability for this class of system, which should be feasible building on our recent work on Lyapunov stability of switched hyperbolic linear systems [PGW12].

Symbolic control of monotone systems - energy management in buildings: Monotone control systems [AS03] constitutes an important class of dynamical systems which have the property of preserving a partial order on the state-space. Similar to incrementally stable systems, symbolic models for monotone control systems are easily computable [MR02]. Most of the applications of monotone control systems are within the field of systems biology. Though, models in several other fields of applications are naturally monotone. For instance, this is the case for thermal dynamics in buildings where models are also often incrementally stable. Based on this observation, we plan to design symbolic control techniques for the purpose of energy management in buildings. The first task to realize is to extend the techniques described in [MR02] in order to handle disturbances, the symbolic models will consists in discrete systems with some controllable and uncontrollable inputs. Then, the controller synthesis problem can be tackled under several angles: centralized or decentralized, robust or stochastic control (depending on the model of disturbances).

Advances in reachability analysis

Reachability techniques for controller synthesis: Controller synthesis for hybrid systems using reachability analysis tries to extend the control techniques originally developed for discrete event systems to the hybrid framework (see e.g. [ABD$^+$00]). For safety properties, it essentially consists in implementing a fixed point algorithm for computing the maximal safety controller. In recent years, there have been spectacular progresses realized in the area of reachability analysis of continuous and hybrid systems. However, most of the attention has been devoted to verification problems rather than synthesis ones (with the exception of [MKM$^+$13]). Building on these state-of-the-art techniques, it is time to revisit these problems and to develop scalable solutions. In my opinion, a problem of particular interest is that of synthesis of guards and reset maps which consists in synthesizing the switching conditions of a hybrid automaton, given the continuous dynamics

and the set of discrete transitions.

Reachability techniques for computation and control co-design: I believe that a promising application of hybrid systems reachability analysis is computation and control co-design, a problem that emerges from research on cyber-physical systems. An approach of particular interest is that of contract based system design [DLTT13]. In this approach, control and software engineers agree on a contract which specifies requirements on the implementation of a controller. Software engineers are responsible for the realization of the implementation, control engineers are responsible for the controller being satisfactory regardless of the implementation respecting these requirements. Hybrid automata constitute the right level of abstraction for studying contract based system design. In particular, given a contract, reachability analysis will allow us to compute all the possible behaviors of the system equipped with any implementation of the controller meeting the requirements. Also, the problem of designing a contract, in order to guarantee some level of performance of the controlled systems can be formulated as a guard synthesis problems for which we plan to develop new approaches. A related problem is event and self-triggered control [HJT12] where the controller is not executed (nearly) periodically but only when some triggering events occur. Most of existing approaches for the design of these events are based on Lyapunov techniques. However, the problem of event generation can also be viewed as a guards and reset maps synthesis problem of a particular hybrid automaton.

Bibliography

[ABC05] T. Alamo, J. M. Bravo, and E. F. Camacho. Guaranteed state estimation by zonotopes. *Automatica*, 41(6):1035–1043, 2005.

[ABD+00] E. Asarin, O. Bournez, T. Dang, O. Maler, and A. Pnueli. Effective synthesis of switching controllers for linear systems. *Proceedings of the IEEE*, 88(7):1011–1025, 2000.

[ABDM00] A. Asarin, O. Bournez, T. Dang, and O. Maler. Approximate reachability analysis of piecewise-linear dynamical systems. In *Hybrid Systems: Computation and Control*, volume 1790 of *LNCS*, pages 20–31. Springer, 2000.

[ACH+95] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995.

[ADG03] E. Asarin, T. Dang, and A. Girard. Reachability analysis of nonlinear systems using conservative approximation. In *Hybrid Systems: Computation and Control*, volume 2623 of *LNCS*, pages 20–35. Springer, 2003.

[ADG07] E. Asarin, T. Dang, and A. Girard. Hybridization methods for the analysis of non-linear systems. *Acta Informatica*, 43(7):451–476, 2007.

[ADI06] R. Alur, T. Dang, and F. Ivancic. Predicate abstraction for reachability analysis of hybrid systems. *ACM Transactions on Embedded Computing Systems*, 5(1):152–199, 2006.

[AHLP00] R. Alur, T. A. Henzinger, G. Lafferriere, and G.J. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(7):971–984, 2000.

[Alu11] R. Alur. Formal verification of hybrid systems. In *International Conference on Embedded Software*, pages 273–278, 2011.

[AMP95] E. Asarin, O. Maler, and A. Pnueli. Reachability analysis of dynamical systems having piecewise-constant derivatives. *Theoretical Computer Science*, 138(1):35–65, 1995.

[Ang02] D. Angeli. A Lyapunov approach to incremental stability properties. *IEEE Transactions on Automatic Control*, 47(3):410–421, 2002.

[AS99] D. Angeli and E. D. Sontag. Forward completeness, unboundedness observability, and their Lyapunov characterizations. *Systems and Control Letters*, 38(3):209–217, 1999.

[AS03] D. Angeli and E. D. Sontag. Monotone control systems. *IEEE Transactions on Automatic Control*, 48(10):1684–1698, 2003.

[ASB07] M. Althoff, O. Stursberg, and M. Buss. Reachability analysis of linear systems with uncertain parameters and inputs. In *IEEE Conference on Decision and Control*, pages 726–732, 2007.

[ASB08] M. Althoff, O. Stursberg, and M. Buss. Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In *IEEE Conference on Decision and Control*, pages 4042–4048, 2008.

[ASB10] M. Althoff, O. Stursberg, and M. Buss. Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes. *Nonlinear Analysis: Hybrid Systems*, 4(2):233–249, 2010.

[ASG00] A. C. Antoulas, D. C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Contemporary Mathematics*, 280:193–219, 2000.

[AT10] A. Anta and P. Tabuada. To sample or not to sample: Self-triggered control for nonlinear systems. *IEEE Transactions on Automatic Control*, 55(9):2030–2042, 2010.

[Aub01] J.-P. Aubin. Viability kernels and capture basins of sets under differential inclusions. *SIAM Journal of Control and Optimization*, 40(3):853–881, 2001.

[BDG+08] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188, 2008.

[Ber00] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2000.

[BFG+93] R. I. Bahar, E. A. Frohm, C. M. Gaona, G. D. Hachtel, E. Macii, A. Pardo, and F. Somenzi. Algebraic decision diagrams and their applications. In *International Conference on Computer-Aided Design*, pages 188–191, 1993.

[BG12a] M. A. Ben Sassi and A. Girard. Computation of polytopic invariants for polynomial dynamical systems using linear programming. *Automatica*, 48(12):3114–3121, 2012.

[BG12b] M. A. Ben Sassi and A. Girard. Controller synthesis for robust invariance of polynomial dynamical systems using linear programming. *Systems and Control Letters*, 61(4):506–512, 2012.

[BGLL08] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communites in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 1742-5468(08):10008+12, 2008.

[BHOT05] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J.N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Joint IEEE Conference Decision and Control and European Control Conference*, pages 2996–3000, 2005.

[BHT09] V. D. Blondel, J. M. Hendrickx, and J. N. Tsitsiklis. On the 2R conjecture for multi-agent systems. *IEEE Transactions on Automatic Control*, 54(11):2506–2517, 2009.

[BPB12] A. Borri, G. Pola, and M. D. Di Benedetto. A symbolic approach to the design of nonlinear networked control systems. In *Hybrid Systems: Computation and Control*, pages 255–264, 2012.

[BT00] O. Botchkarev and S. Tripakis. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In *Hybrid Systems: Computation and Control*, volume 1790 of *LNCS*, pages 73–88. Springer, 2000.

[BTDG12] M. A. Ben Sassi, R. Testylier, T. Dang, and A. Girard. Reachability analysis of polynomial systems using linear programming relaxations. In *Automated Technology for Verification and Analysis*, volume 7561 of *LNCS*, pages 137–151. Springer, 2012.

[CB02] P. Caspi and A. Benveniste. Toward an approximation theory for computerised control. In *International Conference on Embedded Software*, volume 2491 of *LNCS*, pages 294–304. Springer, 2002.

[CG13] A. Colombo and A. Girard. An approximate abstraction approach to safety control of differentially flat systems. In *European Control Conference*, 2013.

[CGG11a] J. Camara, A. Girard, and G. Goessler. Safety controller synthesis for switched systems using multi-scale symbolic models. In *Joint IEEE Conference on Decision and Control and European Control Conference*, pages 520–525, 2011.

[CGG11b] J. Camara, A. Girard, and G. Goessler. Synthesis of switching controllers using approximately bisimilar multiscale abstractions. In *Hybrid Systems: Computation and Control*, pages 191–200, 2011.

[CGP00] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, 2000.

[CK99] A. Chutinan and B. H. Krogh. Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations. In *Hybrid Systems: Computation and Control*, volume 1569 of *LNCS*, pages 76–90. Springer, 1999.

[CL07a] L. Cavarischia and L. Lanari. Hierarchical control implementation. In *Mediterranean Control Conference*, pages 1–6, 2007.

[CL07b] L. Cavarischia and L. Lanari. Hierarchical tracking implementation. In *IEEE Conference on Decision and Control*, pages 3733–3738, 2007.

[Com03] C. Combastel. A state bounding observer based on zonotopes. In *European Control Conference*, 2003.

[CZZ11a] L. Cao, Y. Zheng, and Q. Zhou. A necessary and sufficient condition for consensus of continuous-time agents over undirected time-varying networks. *IEEE Transactions on Automatic Control*, 56(8):1915–1920, 2011.

[CZZ11b] L. Cao, Y. Zheng, and Q. Zhou. A new criterion to linear consensus protocols over time-varying directed networks. In *IEEE International Conference on Control and Automation*, pages 467–470, 2011.

[dAFS04] L. de Alfaro, M. Faella, and M. Stoelinga. Linear and branching metrics for quantitative transition systems. In *International Colloquium on Automata, Languages and Programming*, volume 3142 of *LNCS*, pages 1150–1162. Springer, 2004.

[Dan06] T. Dang. Approximate reachability computation for polynomial systems. In *Hybrid Systems: Computation and Control*, volume 3927 of *LNCS*, pages 138–152. Springer, 2006.

[DCB13] F. Dorfler, M. Chertkov, and F. Bullo. Synchronization in complex oscillator networks and smart grids. *Proceedings of the National Academy of Sciences*, 110(6):2005–2010, 2013.

[DGJP04] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Metrics for labeled Markov processes. *Theoretical Computer Science*, 318(3):323–354, 2004.

[DLTT13] P. Derler, E. A. Lee, M. Torngren, and S. Tripakis. Cyber-physical system design contracts. In *International Conference on Cyber-Physical Systems*, 2013.

[DMT10] T. Dang, O. Maler, and R. Testylier. Accurate hybridization of nonlinear systems. In *Hybrid Systems: Computation and Control*, pages 11–20, 2010.

[DS09] T. Dang and D. Salinas. Image computation for polynomial dynamical systems using the Bernstein expansion. In *Computer Aided Verification*, volume 5643 of *LNCS*, pages 277–287. Springer, 2009.

[EFE06] M. Egerstedt, E. Frazzoli, and G. J. Pappas (Eds). Special section on "symbolic methods for complex control systems". *IEEE Transactions on Automatic Control*, 51(6):921–1013, 2006.

[FGKGP09] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas. Temporal logic planning for dynamic models. *Automatica*, 45(2):343–352, 2009.

[FGP06] G. E. Fainekos, A. Girard, and G. J. Pappas. Temporal logic verification using simulation. In *Formal Modeling and Analysis of Timed Systems*, volume 4202 of *LNCS*, pages 171–186. Springer, 2006.

[FLD+11] G. Frehse, C. Le Guernic, A. Donzé, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. SpaceEx: scalable verification of hybrid systems. In *Computer Aided Verification*, volume 6806 of *LNCS*, pages 379–395. Springer, 2011.

[Fre08] G. Frehse. PHAVer: Algorithmic verification of hybrid systems past HyTech. *Software Tools for Technology Transfer*, 10(3):263–279, 2008.

[FST13] J. Fu, S. Shah, and H. G. Tanner. Hierarchical control via simulation relations and feedback linearization. In *American Control Conference*, 2013.

[GGP09] K. Ghorbal, E. Goubault, and S. Putot. The zonotope abstract domain Taylor1+. In *Computer Aided Verification*, volume 5643 of *LNCS*, pages 627–633. Springer, 2009.

[Gir05] A. Girard. Reachability of uncertain linear systems using zonotopes. In *Hybrid Systems: Computation and Control*, volume 3414 of *LNCS*, pages 291–305. Springer, 2005.

[Gir07] A. Girard. Approximately bisimilar finite abstractions of stable linear systems. In *Hybrid Systems: Computation and Control*, volume 4416 of *LNCS*, pages 231–244. Springer, 2007.

[Gir10] A. Girard. Synthesis using approximately bisimilar abstractions: state-feedback controllers for safety specifications. In *Hybrid Systems: Computation and Control*, pages 111–120, 2010.

[Gir12] A. Girard. Controller synthesis for safety and reachability via approximate bisimulation. *Automatica*, 48(5):947–953, 2012.

[Gir13] A. Girard. Low-complexity quantized switching controllers using approximate bisimulation. *Nonlinear Analysis: Hybrid Systems*, 2013. To appear.

[GJP08] A. Girard, A. A. Julius, and G. J. Pappas. Approximate simulation relations for hybrid systems. *Discrete Event Dynamic Systems*, 18(2):163–179, 2008.

[GL08] A. Girard and C. Le Guernic. Zonotope/hyperplane intersection for hybrid systems reachability analysis. In *Hybrid Systems: Computation and Control*, volume 4981 of *LNCS*, pages 215–228. Springer, 2008.

[GLM06] A. Girard, C. Le Guernic, and O. Maler. Efficient computation of reachable sets of linear time-invariant systems with inputs. In *Hybrid Systems: Computation and Control*, volume 3927 of *LNCS*, pages 257–271. Springer, 2006.

[GM99] M. R. Greenstreet and I. Mitchell. Reachability analysis using polygonal projections. In *Hybrid Systems: Computation and Control*, volume 1569 of *LNCS*, pages 103–116. Springer, 1999.

[GM12] A. Girard and S. Martin. Synthesis for constrained nonlinear systems using hybridization and robust controllers on simplices. *IEEE Transactions on Automatic Control*, 57(4):1046–1051, 2012.

[GP05] A. Girard and G. J. Pappas. Approximate bisimulations for nonlinear dynamical systems. In *Joint IEEE Conference on Decision and Control and European Control Conference*, pages 684–689, 2005.

[GP06] A. Girard and G. J. Pappas. Verification using simulation. In *Hybrid Systems: Computation and Control*, volume 3927 of *LNCS*, pages 272–286. Springer, 2006.

[GP07a] A. Girard and G. J. Pappas. Approximate bisimulation relations for constrained linear systems. *Automatica*, 43(8):1307–1317, 2007.

[GP07b] A. Girard and G. J. Pappas. Approximation metrics for discrete and continuous systems. *IEEE Transactions on Automatic Control*, 52(5):782–798, 2007.

[GP09] A. Girard and G. J. Pappas. Hierarchical control system design using approximate simulation. *Automatica*, 45(2):566–571, 2009.

[GPT10] A. Girard, G. Pola, and P. Tabuada. Approximately bisimilar symbolic models for incrementally stable switched systems. *IEEE Transactions on Automatic Control*, 55(1):116–126, 2010.

[GZ12] A. Girard and G. Zheng. Verification of safety and liveness properties of metric transition systems. *ACM Transactions on Embedded Computing Systems*, 11(S2):54, 2012.

[Hen96] T. A. Henzinger. The theory of hybrid automata. In *IEEE Symposium on Logic in Computer Science*, pages 278–292, 1996.

[HHWT97] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi. HyTech: A model checker for hybrid systems. *Software Tools for Technology Transfer*, 1(1-2):110–122, 1997.

[HHWT98] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi. Algorithmic analysis of nonlinear hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):540–554, 1998.

[HJT12] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada. An introduction to event-triggered and self-triggered control. In *IEEE Conference on Decision and Control*, pages 3270–3285, 2012.

[HK02] R. Hegselmann and U. Krause. Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of Artificial Societies and Social Simulation*, 5(3), 2002.

[HK06] Z. Han and B. H. Krogh. Reachability analysis of large-scale affine systems using low-dimensional polytopes. In *Hybrid Systems: Computation and Control*, volume 3927 of *LNCS*, pages 287–301. Springer, 2006.

[HMP05] T. Henzinger, R. Majumdar, and V. Prabhu. Quantifying similarities between timed systems. In *Formal Modeling and Analysis of Timed Systems*, volume 3829 of *LNCS*, pages 226–241. Springer, 2005.

[HT12] J. M. Hendrickx and J. N. Tsitsiklis. Convergence of type-symmetric and cut-balanced consensus seeking systems. *IEEE Transactions on Automatic Control*, 58(1):214–218, 2012.

[HTP05] E. Haghverdi, P. Tabuada, and G. J. Pappas. Bisimulation relations for dynamical, control, and hybrid systems. *Theoretical Computer Science*, 342(2-3):229–261, 2005.

[HvS01] L. C. G. J. M. Habets and J. H. van Schuppen. Control of piecewise-linear hybrid systems on simplices and rectangles. In *Hybrid Systems: Computation and Control*, volume 2034 of *LNCS*, pages 261–274. Springer, 2001.

[JA10] A. A. Julius and S. Afshari. Using computer games for hybrid systems controller synthesis. In *IEEE Conference on Decision and Control*, pages 5887–5892, 2010.

[JDBP09] A. A. Julius, A. D'Innocenzo, M. D. Di Benedetto, and G. J. Pappas. Approximate equivalence and synchronization of metric transition systems. *Systems and Control Letters*, 58(2):94–101, 2009.

[JFA+07] A. A. Julius, G. Fainekos, M. Anand, I. Lee, and G. J. Pappas. Robust test generation and coverage for hybrid systems. In *Hybrid Systems: Computation and Control*, volume 4416 of *LNCS*, pages 329–342. Springer, 2007.

[JLM03] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[JP09] A. A. Julius and G. J. Pappas. Approximate abstraction of stochastic hybrid systems. *IEEE Transactions on Automatic Control*, 54(6):1193–1203, 2009.

[KB06] M. Kloetzer and C. Belta. A fully automated framework for control of linear systems from LTL specifications. In *Hybrid Systems: Computation and Control*, volume 3927 of *LNCS*, pages 333–347. Springer, 2006.

[Küh98] W. Kühn. Zonotope dynamics in numerical quality control. In *Mathematical Visualization*, pages 125–134. Springer, 1998.

[KV00] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis. In *Hybrid Systems: Computation and Control*, volume 1790 of *LNCS*, pages 202–214. Springer, 2000.

[KV07] A. A. Kurzhanskiy and P. Varaiya. Ellipsoidal techniques for reachability analysis of discrete-time linear systems. *IEEE Transactions on Automatic Control*, 52(1):26–38, 2007.

[LA09] H. Lin and P. J. Antsaklis. Stability and stabilizability of switched linear systems: a survey of recent results. *IEEE Transactions on Automatic Control*, 54(2):308–322, 2009.

[Las01] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[LG09] C. Le Guernic and A. Girard. Reachability analysis of hybrid systems using support functions. In *Computer Aided Verification*, volume 5643 of *LNCS*, pages 540–554. Springer, 2009.

[LG10] C. Le Guernic and A. Girard. Reachability analysis of linear systems using support functions. *Nonlinear Analysis: Hybrid Systems*, 4(2):250–262, 2010.

[Lib03] D. Liberzon. *Switching in systems and control*. Birkhauser, 2003.

[LJS+03] J. Lygeros, K. H. Johansson, S. N. Simic, J. Zhang, and S. S. Sastry. Dynamical properties of hybrid automata. *IEEE Transactions on Automatic Control*, 48(1):2–17, 2003.

[LKCK08] F. Lerda, J. Kapinski, E. M. Clarke, and B. H. Krogh. Verification of supervisory control software using state proximity and merging. In *Hybrid Systems: Computation and Control*, volume 4981 of *LNCS*, pages 344–357. Springer, 2008.

[LS98] W. Lohmiller and J.-J. E. Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, 1998.

[Mal02] O. Maler. Control from computer science. *Annual Reviews in Control*, 26(2):175–187, 2002.

[Mal11] O. Maler. On under-determined dynamical systems. In *International Conference on Embedded Software*, pages 89–96, 2011.

[MEB08] S. Martini, M. Egerstedt, and A. Bicchi. Controllability decompositions of networked systems through quotient graphs. In *IEEE Conference on Decision and Control*, pages 5244–5249, 2008.

[MFJG12] S. Martin, A. Fazeli, A. Jadbabaie, and A. Girard. Multi-agent flocking with random communication radius. In *American Control Conference*, pages 3871–3876, 2012.

[MG10] S. Martin and A. Girard. Sufficent conditions for flocking via graph robustness analysis. In *IEEE Conference on Decision and Control*, pages 6293–6298, 2010.

[MG11] I. C. Morarescu and A. Girard. Opinion dynamics with decaying confidence: application to community detection in graphs. *IEEE Transactions on Automatic Control*, 56(8):1862–1873, 2011.

[MG13] S. Martin and A. Girard. Continuous-time consensus under persistent connectivity and slow divergence of reciprocal interaction weights. *SIAM Journal on Control and Optimization*, 51(3):2568–2584, 2013.

[MGG13] S. Mouelhi, A. Girard, and G. Goessler. CoSyMA: a tool for controller synthesis using multi-scale abstractions. In *Hybrid Systems: Computation and Control*, pages 83–88, 2013.

[MIB+12] A. Murthy, A. Islam, E. Bartocci, E. Cherry, F. H. Fenton, J. Glimm, S.A. Smolka, and R. Grosu. Approximate bisimulations for sodium-channel dynamics. In *Computational Methods in Systems Biology*, volume 7605 of *LNCS*. Springer, 2012.

[Mil89] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[MKM+13] J. N. Maidens, S. Kaynama, I. M. Mitchell, M. M. K. Oishi, and G. A. Dumont. Lagrangian methods for approximating the viability kernel in high-dimensional systems. *Automatica*, 49(7):2017–2029, 2013.

[Mor04] L. Moreau. Stability of continuous-time distributed consensus algorithms. *ArXiv Mathematics e-prints*, September 2004. arXiv:math/0409010.

[Mor05] L. Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, 2005.

[MR99] T. Moor and J. Raisch. Supervisory control of hybrid systems within a behavioral framework. *Systems and Control Letters*, 38(3):157–166, 1999.

[MR02] T. Moor and J. Raisch. Abstraction based supervisory controller synthesis for high order monotone continuous systems. In S. Engell, G. Frehse, and E. Schnieder, editors, *Modelling, Analysis, and Design of Hybrid Systems*, volume 279 of *LNCIS*, pages 247–265. Springer, 2002.

[MT00] I. Mitchell and C. Tomlin. Level set methods for computation in hybrid systems. In *Hybrid Systems: Computation and Control*, volume 1790 of *LNCS*, pages 310–323. Springer, 2000.

[MT10] M. Mazo Jr. and P. Tabuada. Approximate time-optimal control via approximate alternating simulations. In *American Control Conference*, pages 10201–10206, 2010.

[New06] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[NG04] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113.1–15, 2004.

[OSM04] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–2533, 2004.

[Pap03] G. J. Pappas. Bisimilar linear systems. *Automatica*, 39(12):2035–2047, 2003.

[Par03] P.A. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming Ser. B*, 96(2):293–320, 2003.

[PGT08] G. Pola, A. Girard, and P. Tabuada. Approximately bisimilar symbolic models for nonlinear control systems. *Automatica*, 44(10):2508–2516, 2008.

[PGW12] C. Prieur, A. Girard, and E. Witrant. Lyapunov functions for switched linear hyperbolic systems. In *IFAC conference on Analysis and Design of Hybrid Systems*, pages 382–387, 2012.

[PHW03] A. Di Pierro, C. Hankin, and H. Wiklicky. Quantitative relations and approximate process equivalences. In *Conference on Concurrency Theory*, volume 2761 of *LNCS*, pages 508–522. Springer, 2003.

[PPBT10] G. Pola, P. Pepe, M. D. Di Benedetto, and P. Tabuada. Symbolic models for nonlinear time-delay systems using approximate bisimulations. *Systems and Control Letters*, 59(6):365–373, 2010.

[PT07] G. Pola and P. Tabuada. Symbolic models for linear control systems with disturbances. In *IEEE Conference on Decision and Control*, pages 432–437, 2007.

[PT09] G. Pola and P. Tabuada. Symbolic models for nonlinear control systems: alternating approximate bisimulations. *SIAM Journal on Control and Optimization*, 48(2):719–733, 2009.

[PvdSB04] G. Pola, A. J. van der Schaft, and M. D. Di Benedetto. Bisimulation theory for switching linear systems. In *IEEE Conference on Decision and Control*, pages 1406–1411, 2004.

[PVVD09] P. Prabhakar, V. Vladimerou, M. Viswanathan, and G. E. Dullerud. Verifying tolerant systems using polynomial approximations. In *Real-Time Systems Symposium*, pages 181–190, 2009.

[QFD11a] J.-D. Quesel, M. Fränzle, and W. Damm. Crossing the bridge between similar games. In *Formal Modeling and Analysis of Timed Systems*, volume 6919 of *LNCS*, pages 160–176. Springer, 2011.

[QFD11b] J.-D. Quesel, M. Fränzle, and W. Damm. Crossing the bridge between similar games. In *Formal Modeling and Analysis of Timed Systems*, volume 6919 of *LNCS*, pages 160–176, 2011.

[RB05] W. Ren and R. W. Beard. Consensus seeking in multi-agent systems under dynamically changing interaction topologies. *IEEE Transactions on Automatic Control*, 50(5):655–661, 2005.

[RBA07] W. Ren, R. W. Beard, and E. Atkins. Information consensus in multivehicle cooperative control: collective group behavior through local interaction. *IEEE Control Systems Magazine*, 27(2):71–82, 2007.

[Rei09] G. Reißig. Computation of discrete abstractions of arbitrary memory span for nonlinear sampled systems. In *Hybrid Systems: Computation and Control*, volume 5469 of *LNCS*, pages 306–320. Springer, 2009.

[RMC10] N. Ramdani, N. Meslem, and Y. Candau. Computing reachable sets for uncertain nonlinear monotone systems. *Nonlinear Analysis: Hybrid Systems*, 4(2):263–278, 2010.

[RO98] J. Raisch and S. O'Young. Discrete approximation and supervisory control of continuous systems. *IEEE Transactions on Automatic Control*, 43(4):569–573, 1998.

[RW87] P. J. Ramadge and W. M. Wonham. Modular feedback logic for discrete event systems. *SIAM Journal on Control and Optimization*, 25(5):1202–1218, 1987.

[SB13] Y. Seladji and O. Bouissou. Numerical abstract domain using support functions. In *NASA Formal Methods*, volume 7871 of *LNCS*, pages 155–169. Springer, 2013.

[SK03] O. Stursberg and B. H. Krogh. Efficient representation and computation of reachable sets for hybrid systems. In *Hybrid Systems: Computation and Control*, volume 2623 of *LNCS*, pages 482–497. Springer, 2003.

[SP94] P. Saint-Pierre. Approximation of the viability kernel. *Applied Mathematics and Optimization*, 29:187–209, 1994.

[Tab06] P. Tabuada. Symbolic control of linear systems based on symbolic subsystems. *IEEE Transactions on Automatic Control*, 51(6):1003–1013, 2006.

[Tab08] P. Tabuada. An approximate simulation approach to symbolic control. *IEEE Transactions on Automatic Control*, 53(6):1406–1418, 2008.

[Tab09] P. Tabuada. *Verification and Control of Hybrid Systems - A Symbolic Approach*. Springer, 2009.

[TAJP08] P. Tabuada, A. D. Ames, A. Julius, and G. J. Pappas. Approximate reduction of dynamic systems. *Systems and Control Letters*, 57(7):538–545, 2008.

[TI08] Y. Tazaki and J. I. Imura. Finite abstractions of discrete-time linear systems and its application to optimal control. In *IFAC World Congress*, pages 10201–10206, 2008.

[TI09] Y. Tazaki and J. I. Imura. Discrete-state abstractions of nonlinear systems using multi-resolution quantizer. In *Hybrid Systems: Computation and Control*, volume 5469 of *LNCS*, pages 351–365. Springer, 2009.

[TMBO03] C. Tomlin, I. Mitchell, A. M. Bayen, and M. Oishi. Computational techniques for the verification of hybrid systems. *Proceedings of the IEEE*, 91(7):986–1001, 2003.

[TP06] P. Tabuada and G. J. Pappas. Linear time logic control of discrete-time linear systems. *IEEE Transactions on Automatic Control*, 51(12):1862–1877, 2006.

[vB05] F. van Breugel. A behavioural pseudometric for metric labelled transition systems. In *Conference on Concurrency Theory*, volume 3653 of *LNCS*, pages 141–155. Springer, 2005.

[vBMOW03] F. van Breugel, M. Mislove, J. Ouaknine, and J. Worrell. An intrinsic characterization of approximate probabilistic bisimilarity. In *Foundations of Software Science and Computation Structures*, volume 2620 of *LNCS*, pages 200–215. Springer, 2003.

[vdS04] A. J. van der Schaft. Equivalence of dynamical systems by bisimulation. *IEEE Transactions on Automatic Control*, 49(12):2160–2172, 2004.

[VFM03] M. Velasco, J. Fuertes, and P. Marti. The self triggered task model for real-time control systems. In *IEEE Real-Time Systems Symposium*, pages 67–70, 2003.

[YW00] M. Ying and M. Wirsing. Approximate bisimilarity. In *Algebraic Methodology and Software Technology*, volume 1816 of *LNCS*, pages 309–322, 2000.

[Zie95] G. M. Ziegler. *Lectures on polytopes*. Springer-Verlag, 1995.

[ZMM+13] M. Zamani, P. Mohajerin Esfahani, R. Majumdar, A. Abate, and J. Lygeros. Symbolic control of stochastic systems via approximately bisimilar finite abstractions. *ArXiv e-prints*, February 2013. arXiv:1302.3868.

[ZPJT12] M. Zamani, G. Pola, M. Mazo Jr., and P. Tabuada. Symbolic models for nonlinear control systems without stability assumptions. *IEEE Transactions on Automatic Control*, 57(7):1804–1809, 2012.

List of Publications

Peer-Reviewed International Journals:

J20. Samuel Martin et Antoine Girard, Continuous-time consensus under persistent connectivity and slow divergence of reciprocal interaction weights. SIAM Journal on Control and Optimization, 51(3):2568-2584, 2013.

J19. Antoine Girard, Low-complexity quantized switching controllers using approximate bisimulation. Nonlinear Analysis: Hybrid Systems, 10:34-44, 2013.

J18. Mohamed Amin Ben Sassi et Antoine Girard, Computation of polytopic invariants for polynomial dynamical systems using linear programming. Automatica, 48(12):3114-3121, 2012.

J17. Antoine Girard et Gang Zheng, Verification of safety and liveness properties of metric transition systems. ACM Transactions on Embedded Computing Systems, special issue on Verification of Cyber-Physical Software Systems, 11(S2), Article No. 54, 2012.

J16. Truong Nghiem, George J. Pappas, Rajeev Alur et Antoine Girard, Time-triggered implementations of dynamic controllers. ACM Transactions on Embedded Computing Systems, special issue on Verification of Cyber-Physical Software Systems, 11(S2), Article No. 58, 2012.

J15. Antoine Girard, Controller synthesis for safety and reachability via approximate bisimulation. Automatica, 48(5):947:953, 2012.

J14. Antoine Girard et Samuel Martin, Synthesis for constrained nonlinear systems using hybridization and robust controllers on simplices. IEEE Transactions on Automatic Control, 57(4):1046-1051, 2012.

J13. Mohamed Amin Ben Sassi et Antoine Girard, Controller synthesis for robust invariance of polynomial dynamical systems using linear programming. Systems and Control Letters, 61(4):506-512, 2012.

J12. Antoine Girard et George J. Pappas, Approximate bisimulation: a bridge between computer science and control theory. European Journal of Control, 17(5-6):568-578, 2011.

J11. Irinel Constantin Morarescu et Antoine Girard, Opinion dynamics with decaying confidence: application to community detection in graphs. IEEE Transactions on Automatic Control, 56(8):1862-1873, 2011.

J10. Colas Le Guernic et Antoine Girard, Reachability analysis of linear systems using support functions. Nonlinear Analysis: Hybrid Systems, special issue related to the 2008 IFAC World Congress, 4(2):250-262, 2010.

J9. Antoine Girard, Giordano Pola et Paulo Tabuada, Approximately bisimilar symbolic models for incrementally stable switched systems. IEEE Transactions on Automatic Control, 55(1):116-126, 2010.

J8. Antoine Girard et George J. Pappas, Hierarchical control system design using approximate simulation. Automatica, 45(2):566-571, 2009.

J7. Georgios E. Fainekos, Antoine Girard, Hadas Kress-Gazit et George J. Pappas, Temporal logic planning for dynamic models. Automatica, 45(2):343-352, 2009.

J6. Giordano Pola, Antoine Girard et Paulo Tabuada, Approximately bisimilar symbolic models for nonlinear control systems. Automatica, 44(10):2508-2516, 2008.

J5. Antoine Girard, A. Agung Julius et George J. Pappas, Approximate simulation relations for hybrid systems. Discrete Event Dynamic Systems, special issue on Discrete Event Methodologies for Hybrid Systems, 18(2):163-179, 2008.

J4. Antoine Girard et George J. Pappas, Approximate bisimulation relations for constrained linear systems. Automatica, 43(8):1307-1317, 2007.

J3. Antoine Girard et George J. Pappas, Approximation metrics for discrete and continuous systems. IEEE Transactions on Automatic Control, 52(5):782-798, 2007.
2009 George S. Axelby Outstanding Paper Award.

J2. Eugène Asarin, Thao Dang et Antoine Girard, Hybridization methods for the analysis of non-linear systems. Acta Informatica, special issue on Hybrid Systems, 43(7):451-476, 2007.

J1. Antoine Girard, Towards a multiresolution approach to linear control. IEEE Transactions on Automatic Control, 51(8):1261-1270, 2006.

Peer-Reviewed International Conference Proceedings:

C49. Pierre-Jean Meyer, Antoine Girard et Emmanuel Witrant, Controllability and invariance of monotone systems for robust ventilation automation in buildings. IEEE Conference on Decision and Control, Florence, Italie, décembre 2013.

C48. Euriell Le Corronc, Antoine Girard et Gregor Goessler, Mode sequences as symbolic states in abstractions of incrementally stable switched systems. IEEE Conference on Decision and Control, Florence, Italie, décembre 2013.

C47. Ying Tang, Christophe Prieur et Antoine Girard, A new H2-norm Lyapunov function for the stability of a singularly perturbed system of two conservation laws. IEEE Conference on Decision and Control, Florence, Italie, décembre 2013.

C46. Pierre-Olivier Lamare, Antoine Girard et Christophe Prieur, Lyapunov techniques for stabilization of switched linear systems of conservation laws. IEEE Conference on Decision and Control, Florence, Italie, décembre 2013.

C45. Ying Tang, Christophe Prieur et Antoine Girard, Lyapunov stability of a singularly perturbed system of two conservation laws. IFAC Workshop on the Control of Systems Modeled by Partial Differential Equations, Paris, France, septembre 2013.

C44. Alessandro Colombo et Antoine Girard, An approximate abstraction approach to safety control of differentially flat systems. European Control Conference, Zurich, Suisse, juillet 2013.

C43. Mohamed Amin Ben Sassi et Antoine Girard, Control of polynomial dynamical systems on rectangles. European Control Conference, Zurich, Suisse, juillet 2013.

C42. Sebti Mouelhi, Antoine Girard et Gregor Goessler, CoSyMA: a tool for controller synthesis using multi-scale abstractions. Hybrid Systems: Computation and Control, Philadelphie, USA, avril 2013.

C41. Takayuki Ishizaki, Kenji Kashima, Antoine Girard, Jun-ichi Imura, Luonan Chen and Kazuyuki Aihara, Clustering-based H2-state aggregation of positive networks and its application to reduction of chemical master equations. IEEE Conference on Decision and Control, Maui, USA, décembre 2012.

C40. Mohamed Amin Ben Sassi, Romain Testylier, Thao Dang et Antoine Girard, Reachability analysis of polynomial systems using linear programming relaxations. Automated Technology for Verification and Analysis, Thiruvananthapuram, India, 2012.

C39. Christophe Prieur, Antoine Girard et Emmanuel Witrant, Lyapunov functions for switched linear hyperbolic systems. 4th IFAC conference on Analysis and Design of Hybrid Systems, Eindhoven, Pays-Bas, juin 2012.

C38. Antoine Girard, Low-complexity switching controllers for safety using symbolic models. 4th IFAC conference on Analysis and Design of Hybrid Systems, Eindhoven, Pays-Bas, juin 2012.

C37. Samuel Martin, Arastoo Fazeli, Ali Jadbabaie et Antoine Girard, Multi-agent flocking with random communication radius. American Control Conference, Montreal, Canada, juillet 2012.

C36. Javier Camara, Antoine Girard et Gregor Goessler, Safety controller synthesis for switched systems using multi-scale symbolic models. Joint IEEE Conference on Decision and Control and European Control Conference, Orlando, USA, décembre 2011.

C35. Goran Frehse, Colas Le Guernic, Alexandre Donzé, Rajarshi Ray, Olivier Lebeltel, Rodolfo Ripado, Antoine Girard, Thao Dang et Oded Maler, SpaceEx: scalable verification of hybrid systems. Computer Aided Verification, Snowbird, USA, juillet 2011.

C34. Javier Camara, Antoine Girard et Gregor Goessler, Synthesis of switching controllers using approximately bisimilar multiscale abstractions. Hybrid Systems: Computation and Control, Chicago, USA, avril 2011.

C33. Samuel Martin et Antoine Girard, Sufficent conditions for flocking via graph robustness analysis. IEEE Conference on Decision and Control, Atlanta, USA, décembre 2010.

C32. Antoine Girard, Synthesis using approximately bisimilar abstractions: time-optimal control problems. IEEE Conference on Decision and Control, Atlanta, USA, décembre 2010.

C31. I. Constantin Morarescu et Antoine Girard, Consensus with constrained convergence rate: agreement in communities. IEEE Conference on Decision and Control, Atlanta, USA, décembre 2010.

C30. I. Constantin Morarescu et Antoine Girard, A model of opinion dynamics for community detection in graphs. IFAC Workshop on Distributed Estimation and Control in Networked Systems, Annecy, France, septembre 2010.

C29. I. Constantin Morarescu, Silviu I. Niculescu, Antoine Girard, Consensus with constrained convergence rate and time-delays. 9th IFAC Workshop on Time Delay Systems, Prague, République Tchèque, juin 2010.

C28. Antoine Girard, Synthesis using approximately bisimilar abstractions: state-feedback controllers for safety specifications. Hybrid Systems: Computation and Control, pp 111-120, Stockholm, Suède, avril 2010.

C27. Colas Le Guernic et Antoine Girard, Reachability analysis of hybrid systems using support functions. Computer Aided Verification, vol 5643/2009 in LNCS, pp 540-554, Springer, Grenoble, France, juin 2009.

C26. Gang Zheng et Antoine Girard, Bounded and unbounded safety verification using bisimulation metrics. Hybrid Systems: Computation and Control, vol 5469/2009 in LNCS, pp 426-440, Springer, San Francisco, USA, avril 2009.

C25. Antoine Girard et Samuel Martin, Motion planning for nonlinear systems using hybridizations and robust controllers on simplices. 47th IEEE Conference on Decision and Control, pp 239 - 244, Cancun, Mexique, décembre 2008.

C24. Antoine Girard et Colas Le Guernic, Efficient reachability analysis for linear systems using support functions. IFAC World Congress, Séoul, Corée du Sud, juillet 2008.

C23. Antoine Girard et Colas Le Guernic, Zonotope/hyperplane intersection for hybrid systems reachability analysis. Hybrid Systems: Computation and Control, vol 4981 in LNCS, pp 215-228, Springer, Saint-Louis, USA, avril 2008.

C22. Antoine Girard, Giordano Pola et Paulo Tabuada, Approximately bisimilar symbolic models for incrementally stable switched systems. Hybrid Systems: Computation and Control, vol 4981 in LNCS, pp 201-214, Springer, Saint-Louis, USA, avril 2008.

C21. Giordano Pola, Antoine Girard et Paulo Tabuada, Symbolic models for nonlinear control systems using approximate bisimulation. 46th IEEE Conference on Decision and Control, pp 4656-4661, New Orleans, USA, décembre 2007.

C20. Antoine Girard et George J. Pappas, Approximate hierarchies of linear control systems. 46th IEEE Conference on Decision and Control, pp 3727-3732, New Orleans, USA, décembre 2007.

C19. Georgios E. Fainekos, Antoine Girard et George J. Pappas, Hierarchical synthesis of hybrid controllers from temporal logics specifications. Hybrid Systems: Computation and Control, vol 4416 in LNCS, pp 203-216, Springer, Pise, Italie, avril 2007.

C18. Antoine Girard, Approximately bisimilar finite abstractions of stable linear systems. Hybrid Systems: Computation and Control, vol 4416 in LNCS, pp 231-244, Springer, Pise, Italie, avril 2007.

C17. Antoine Girard et George J. Pappas, Hierarchical control using approximate simulation relations. 45th IEEE Conference on Decision and Control, San Diego, USA, décembre 2006.

C16. Truong Nghiem, George J. Pappas, Antoine Girard et Rajeev Alur, Time-triggered implementations of dynamic controllers. 6th ACM and IEEE Conference on Embedded Software, Séoul, Corée du Sud, octobre 2006.

C15. Eugène Asarin, Thao Dang, Goran Frehse, Antoine Girard, Colas Le Guernic et Oded Maler, Recent progress in continuous and hybrid reachability analysis. IEEE International Symposium on Computer-Aided Control Systems Design, Munich, Allemagne, octobre 2006.

C14. Georgios E. Fainekos, Antoine Girard et George J. Pappas, Temporal logic verification using simulation. Formal Modelling and Analysis of Timed Systems, LNCS 4202, pp 171-186, Springer, Paris, France, septembre 2006.

C13. Antoine Girard, A. Agung Julius et George J. Pappas, Approximate simulation relations for hybrid systems. 2nd IFAC Conference on Analysis and Design of Hybrid Systems, pp 106-111, Alghero, Italie, juin 2006.

C12. A. Agung Julius, Antoine Girard et George J. Pappas, Approximate bisimulation for a class of stochastic hybrid systems. American Control Conference, Portland, USA, juin 2006.

C11. Antoine Girard et George J. Pappas, Verification using simulation. Hybrid Systems : Computation and Control, LNCS 3927, pp 272-286, Springer, Santa-Barbara, USA, mars 2006.

C10. Antoine Girard, Colas Le Guernic et Oded Maler, Efficient computation of reachable sets of linear time-invariant systems with inputs. In Hybrid Systems: Computation and Control, LNCS 3927, pp 257-271, Springer, Santa-Barbara, USA, mars 2006.

C9. Antoine Girard et George J. Pappas, Approximate bisimulations for constrained linear systems. 44th IEEE Conference on Decision and Control and European Control Conference, pp 4700-4705, Séville, Espagne, décembre 2005.

C8. Antoine Girard et George J. Pappas, Approximate bisimulations for nonlinear dynamical systems. 44th IEEE Conference on Decision and Control and European Control Conference, pp 684-689, Séville, Espagne, décembre 2005.

C7. Hakan Yazarel, Antoine Girard, George J. Pappas et Rajeev Alur, Quantifying the gap between embedded control models and time-triggered implementations. 26th IEEE Real-Time Systems Symposium, pp 111-120, Miami, USA, décembre 2005.

C6. Antoine Girard, Reachability of uncertain linear systems using zonotopes. Hybrid Systems: Computation and Control, LNCS 3414, pp 291-305, Springer, Zurich, Suisse, mars 2005.

C5. Antoine Girard, Optimal control of linear systems: a multiresolution approach. 43rd IEEE Conference on Decision and Control, pp 1806-1811, Nassau, Bahamas, décembre 2004.

C4. Antoine Girard, Computation and stability analysis of limit cycles in piecewise linear hybrid systems. 1st IFAC Conference on Analysis and Design of Hybrid Systems, pp 181-186, Saint-Malo, France, Juin 2003.

C3. Eugène Asarin, Thao Dang et Antoine Girard, Reachability analysis of non-linear systems using conservative approximations. Hybrid Systems: Computation and Control, vol 2623 in LNCS, pp 20-35, Springer, Prague, République Tchèque, 2003.

C2. Antoine Girard, Detection of event occurrence in piecewise linear hybrid systems. 4th International Conference on Recent Advances in Soft Computing, pp 19-25, Nottingham, Royaume Uni, Décembre 2002.

C1. Antoine Girard, Approximate solutions of ODEs using piecewise linear vector fields. 5th International Workshop on Computer Algebra in Scientific Computing, pp 107-120, Yalta, Ukraine, Septembre 2002.

Book Chapters:

CL2. Thao Dang, Goran Frehse, Antoine Girard et Colas Le Guernic, Tools for the analysis of hybrid models. Dans Claude Jard et Olivier H. Roux, éditeurs, *Communicating Embedded Systems - Software and Design*. ISTE Publishing/John Wiley, 2009.

CL1. Thao Dang, Goran Frehse, Antoine Girard et Colas Le Guernic, Outils pour l'analyse des modèles hybrides. Dans Claude Jard et Olivier H. Roux, éditeurs, *Approches formelles des systèmes embarqués communicants*. Traité IC2, série Informatique et systèmes d'information, Hermes, 2008.

Thesis:

M2. Antoine Girard, Analyse algorithmique des systèmes hybrides. Thèse de doctorat, Institut National Polytechnique de Grenoble, France, septembre 2004.

M1. Antoine Girard, Etude de systèmes dynamiques hybrides affines par morceaux. Mémoire de DEA, Université Joseph Fourier, Grenoble, France, juin 2001.

Résumé

Un système hybride est un système dynamique exhibant à la fois des comportements de nature discrète et continue. Motivée par la multiplication de composants informatiques embarqués "discrets" interagissant avec le monde physique "continu", la recherche sur les systèmes hybrides s'est développée rapidement depuis les années 90 à l'intersection de l'informatique, de l'automatique et des mathématiques appliquées. Ce mémoire présente nos contributions, théoriques ou méthodologiques, à ce domaine. Dans une première partie, nous introduisons un cadre d'approximation qui s'applique aux systèmes dynamiques continus, discrets et hybrides; des applications, notamment dans le domaine du contrôle symbolique sont présentées. La deuxième partie est consacrée à l'analyse d'atteignabilité, une technique computationnelle très utile pour l'analyse des systèmes hybrides. Enfin, la troisième partie porte sur les systèmes dynamiques multi-agents.

Abstract

A hybrid system is a dynamical system exhibiting both continuous and discrete behaviors. Motivated by the multiplication of "discrete" embedded computing devices interacting with the "continuous" physical world, the research on hybrid systems has rapidly developed since the nineties at the intersection of computer science, control theory and applied mathematics. This thesis presents our theoretical and methodological contributions to this field. In a first part, we introduce an approximation framework that applies to dynamical systems that can be continuous, discrete or hybrid; applications, including some in the field of symbolic control, are presented. The second part deals with reachability analysis, a computational technique which is very useful for the analysis of hybrid systems. Finally, the third part presents our contributions to multi-agent dynamical systems.