UNIVERSITÉ GRENOBLE ALPES

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Automatique-Productique**

Arrêté ministériel : 7 août 2006

Présentée par

Pierre-Jean MEYER

Thèse dirigée par **Antoine GIRARD**
et par **Emmanuel WITRANT**

préparée au sein du **Laboratoire Jean Kuntzmann (LJK)**
dans l'école doctorale **Electronique, Electrotechnique, Automatique et Traitement du Signal (EEATS)**

Invariance and symbolic control of cooperative systems for temperature regulation in intelligent buildings

Thèse soutenue publiquement le **24 septembre 2015**
devant le jury composé de :

M. Nacim RAMDANI
Professeur, Université d'Orléans, Rapporteur

M. Paulo TABUADA
Professeur, University of California, Rapporteur

M. Mazen ALAMIR
Directeur de recherche CNRS, Université Grenoble Alpes, Examinateur

M. Jean-Luc GOUZE
Directeur de recherche INRIA, INRIA Sophia-Antipolis, Examinateur

M. Hervé GUEGUEN
Professeur, Supélec Rennes, Examinateur

M. Antoine GIRARD
Maître de conférences, Université Grenoble Alpes, Directeur de thèse

M. Emmanuel WITRANT
Maître de conférences, Université Grenoble Alpes, Directeur de thèse

Université Grenoble Alpes

Ecole doctorale EEATS

Invariance and symbolic control of cooperative systems for temperature regulation in intelligent buildings

Pierre-Jean Meyer

Thesis prepared under the supervision of:

Antoine Girard
in Laboratoire Jean Kuntzmann (LJK), Grenoble, France

and

Emmanuel Witrant
in GIPSA-lab, Grenoble, France

September 24, 2015

Remerciements

First of all, I would like to thank Nacim Ramdani and Paulo Tabuada for accepting and taking the time to review the work of my thesis in details and synthesizing it into these particularly positive reviews that I received. Je tiens également à remercier Mazen Alamir, Jean-Luc Gouzé et Hervé Guéguen pour avoir accepté de faire partie de mon jury de thèse.

C'est vers Antoine et Emmanuel que je me tourne pour adresser mes principaux remerciements : c'est grâce à vous que ce très intéressant sujet de thèse a vu le jour et je vous remercie de m'avoir fait confiance pour mener à bien ce projet. Au cours de ces trois années de thèse, vous avez su me guider dans mes travaux tout en me laissant une grande autonomie. Nos interactions fréquentes et votre grande disponibilité a permis de créer un environnement de travail qui m'a été idéal. Je tiens aussi à vous remercier pour m'avoir incité à communiquer et publier nos résultats auprès des communautés française et internationale à de nombreuses reprises et d'avoir supporté financièrement le grand nombre de déplacements professionnels que cela impliquait. De mon point de vue, cette thèse a été une formidable expérience, à la fois sur les aspects scientifiques qu'humains, et vous en êtes les principaux responsables. J'espère que mon séjour à vos côtés vous aura laissé une aussi bonne impression. Je vous remercie enfin pour votre soutien allant bien au-delà de la thèse, entre autre lors de mes recherches de post-doc.

Je remercie la région Rhône-Alpes pour avoir partiellement financé ma thèse dans le cadre du projet COHYBA (contrôle hybride pour les bâtiments verts). I would also like to thank Hosein for his internship and acknowledge how important his work on the experimental building has been for my thesis. You saved me a lot of time and worries by making this pile of PVC and electronics into a working experiment. Merci également à Marie-Pierre Vaillant et à l'équipe technique de PhITEM pour leur assistance sur les problèmes matériels et électriques.

In the middle of my thesis, I had the great opportunity to visit the Department of Automatic Control in KTH and I would like to thank Kalle Johansson and his team for welcoming me. I am particularly grateful for the interactions I had with Alessandra, Marco, Afrooz, Giulio, Damiano, Lin, Behdad, Luca, Jana, Assad, Davide, Bart and Dimos.

La vie au LJK est belle, et elle est d'autant facilitée par la disponibilité et l'efficacité des services informatique et administratif. Je remercie tout particulièrement

Laurence, Delphine, Juana, Fred, Franck et Cathy avec qui j'ai eu le plus d'interactions.

L'expérience de la thèse n'aurait pas été complète sans une pléthore de soirées jeux (voire des nuits complètes), mots fléchés, thés des doctorants, bières, randonnées (de préférence nocturnes et avec des crêpes), assauts d'escrime ou matchs de foot et d'ultimate. Un très grand merci à tous les doctorants et post-doctorants du LJK avec qui j'ai pu vivre toutes ces aventures et tout particulièrement à Vincent, Romain, Morgane, Pierre-O, Matthias, Cécile, Chloé, Nelson, Meriem, Rémi, Roland, Madison, Bertrand, Kévin, Ester, Jean-Matthieu, Burak, JB, Euriell et Thomas. La liste est bien sûr loin d'être exhaustive et je tiens tout de même à remercier tous mes autres amis du LJK qui ont aussi fortement participé à rendre mémorable ces 3 années au LJK.

Les amis du GIPSA ont beau être moins nombreux, Nicolás, Swann, Peter, Bojan et Ying ont aussi droit à leur petit paragraphe !

Merci également à mes partenaires d'escrime, que ce soit sur le campus ou au club de Seyssins, pour m'avoir laissé me défouler sur eux, et merci tout particulièrement à Maître Michel pour m'y avoir accueilli.

Terminons enfin ces remerciements sur ceux que j'adresse à Pierre, Catherine, Florent et Lisa, sans raison particulière, juste parce que j'en ai envie.

Contents

Résumé de la thèse

Motivations

Bâtiments verts Etant données la forte croissance de la population mondiale et l'augmentation en termes de demande de confort, l'efficacité énergétique dans les bâtiments est devenue une préoccupation majeure au niveau mondial, et plus particulièrement dans les pays développés où les bâtiments peuvent représenter jusqu'à 40% de la consommation d'énergie [PLOP08]. Il a été montré que la coordination des décisions de contrôle liées à la régulation du climat intérieur (température, ventilation, éclairage, humidité, ...) pouvait réduire la consommation d'énergie globale du bâtiment de manière significative. Ce type d'approche nécessite l'introduction d'éléments de mesure, de coordination et d'action dans le bâtiment pour pouvoir connaître la situation actuelle, déterminer la stratégie de contrôle la plus efficace au niveau global du bâtiment et mettre en œuvre cette stratégie localement dans chacune des pièces. L'utilisation de ces composants technologiques et décisionnels dans un bâtiment correspond à la description de base d'un bâtiment énergétiquement efficace, aussi connu sous le nom de bâtiment intelligent ou bâtiment vert.

Dans ce travail, nous nous intéressons à la régulation de la température dans un bâtiment vert. C'est un problème compliqué du fait de l'hétérogénéité des éléments influençant le comportement global. En effet, ce type de systèmes exhibe à la fois des comportements continus (variations de la température selon la première loi de la thermodynamique) et des transitions discrètes (par exemple, un utilisateur qui entre dans une pièce ou qui ouvre une fenêtre) et ne peuvent donc être correctement décrits que dans le cadre mathématique des systèmes hybrides. Du fait de ces interactions, nous ne pouvons pas appliquer les méthodes classiques venant des théories du contrôle pour les systèmes continus ou pour les systèmes discrets et il nous faut donc utiliser des techniques spécifiques adaptées à la nature hybride du système.

Contrôle symbolique La solution étudiée dans cette thèse pour résoudre ce problème de contrôle est basée sur des méthodes symboliques. Le principe de ces méthodes est de créer une abstraction purement discrète du système original que l'on représentera sous la forme d'un système de transitions fini et non-déterministe pour lequel un contrôleur est plus facile à synthétiser grâce aux méthodes dans le domaine du contrôle discret. Si une relation comportementale (simulation, bisimulation, ou leurs versions alternées et approchées [Tab09]) entre l'abstraction et le modèle original peut être prouvée, cela signifie que tous comportements du système original peuvent être reproduits dans l'abstraction. La relation de simulation alternée

implique également qu'un contrôleur discret synthétisé sur l'abstraction peut être transformé en un contrôleur du modèle original satisfaisant les mêmes spécifications. Nous parlons ainsi de *contrôle hybride* puisqu'un contrôleur discret est appliqué à un système continu (ou hybride). Il faut noter que ce nom ne veut pas forcément dire que cette approche ne s'utilise que pour les systèmes hybrides : elle peut être intéressante pour tous systèmes dont les dynamiques sont trop complexes pour être contrôlées avec les méthodes classiques.

Le nom de méthodes *symboliques* s'explique par la première étape de la création de l'abstraction discrète, consistant en une partition de l'espace d'état : chaque élément de cette partition peut être vu comme un *symbole* représentant tous les états continus qu'il contient. Les transitions de l'abstraction symbolique sont ensuite obtenues à l'aide d'une analyse d'atteignabilité pour laquelle on prend une approximation de l'ensemble des états continus qui peuvent être atteints (avec une version échantillonnée du système de départ) à partir de ceux contenus dans un symbole. Cette approximation peut être déterminée de plusieurs manières selon les propriétés du système, mais la méthode la plus simple peut-être utilisée lorsque le système satisfait une propriété de monotonie, décrite dans le paragraphe suivant.

Monotonie Les systèmes possédant la propriété de monotonie apparaissent dans une grande variété de domaines tels que la biologie moléculaire, les réseaux biochimiques, les évolutions de population ou les dynamiques thermiques dans les bâtiments. Un système monotone est défini comme un système dont les trajectoires préservent un ordre partiel sur ses états [AS03]. Cela signifie que si l'on considère un état initial x_0 *plus grand* qu'un autre x'_0 et une fonction d'entrée $\mathbf{u}$ à tout instant *plus grande* qu'une autre $\mathbf{u}'$, alors la trajectoire du système initialisé en x_0 avec l'entrée $\mathbf{u}$ reste toujours *au dessus* de la trajectoire initialisée en x'_0 avec l'entrée $\mathbf{u}'$. La sous-classe des systèmes coopératifs correspond au cas particulier où les ordres partiels choisis sont les inégalités classiques sur chaque composante des variables vectorielles comparées :

$$x_0 \geq x'_0,\ \forall t \in \mathbb{R}_0^+,\ \mathbf{u}(t) \geq \mathbf{u}'(t) \Rightarrow \forall t \in \mathbb{R}_0^+,\ \mathbf{x}(t; x_0, \mathbf{u}) \geq \mathbf{x}(t; x'_0, \mathbf{u}').$$

Cette propriété est particulièrement utile pour borner n'importe quelle trajectoire du système par deux trajectoires particulières impliquant les valeurs extrémales (à l'égard de l'ordre partiel choisi) de l'état et des variables d'entrées. Ainsi, pour la création de l'abstraction symbolique, si les symboles choisis ont la forme d'intervalles (à plusieurs dimensions) de l'espace d'état, une sur-approximation de l'ensemble atteignable peut être obtenue en ne calculant que deux successeurs du système échantillonné : un pour la borne inférieure du symbole considéré et un pour sa borne supérieure.

Invariance contrôlée Pour le problème de régulation de la température dans un bâtiment, chaque utilisateur choisit une température de référence correspondant à ses critères de confort pour la pièce qu'il occupe. Puisque nous considérons un système subissant des perturbations inconnues mais bornées, la notion classique de stabilité peut ne pas être satisfaite et il nous faut donc relâcher les spécifications de confort en utilisant plutôt des intervalles de températures autour des valeurs de

référence demandées. Par conséquent, répondre aux spécifications globales en termes de confort revient à trouver une stratégie de contrôle maintenant l'état du système (le vecteur des températures du bâtiment) dans un intervalle multi-dimensionnel malgré le comportement antagoniste de l'environnement. Dans ces travaux, il est fait référence à cette notion sous le nom de *jeu de sûreté* pour les systèmes évoluant en temps discret (comme c'est le cas pour l'abstraction symbolique) et sous le nom d'*invariance contrôlée robuste* pour les systèmes en temps continu.

Puisque cet objectif d'invariance est dans un intervalle vectoriel qui possède naturellement une borne inférieure et une borne supérieure, la propriété de monotonie peut aussi s'avérer utile pour caractériser la notion d'invariance contrôlée robuste. Bien que cette caractérisation décrive la capacité de contrôler le système dans un ensemble plutôt que de fournir une véritable stratégie de contrôle, elle donne des éléments de comparaison intéressants avec les méthodes symboliques et facilite le choix de l'intervalle à considérer dans l'étape d'abstraction.

Chapitre 1 : Systèmes coopératifs

Dans cette thèse, nous considérons le système dynamique suivant, évoluant en temps continu :

$$\dot{x} = f(x, u, w), \tag{1.1}$$

où $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$ et $w \in \mathbb{R}^q$ sont l'état, l'entrée de contrôle et l'entrée de perturbation, respectivement. Les trajectoires de (1.1) sont décrites par $\Phi(\cdot, x_0, \mathbf{u}, \mathbf{w})$, où $\Phi(t, x_0, \mathbf{u}, \mathbf{w})$ représente l'état atteint au temps $t \in \mathbb{R}_0^+$ à partir de l'état initial $x_0 \in \mathbb{R}^n$ et avec les fonctions de contrôle et de perturbation $\mathbf{u} : \mathbb{R}_0^+ \to \mathbb{R}^p$ et $\mathbf{w} : \mathbb{R}_0^+ \to \mathbb{R}^q$.

Dans la suite, l'inégalité $\geq$ utilisée pour comparer des vecteurs correspond à l'inégalité classique sur chaque composantes. Cette notation est également utilisée pour les fonctions du temps pour lesquelles l'inégalité doit être satisfaite à tout instant : $\mathbf{u} \geq \mathbf{u}' \Leftrightarrow \forall t \geq 0,\ \mathbf{u}(t) \geq \mathbf{u}'(t)$. Le système décrit par (1.1) est dit *coopératif* si ses trajectoires préservent ces inégalités [AS03].

Définition 1.5 (Système coopératif)**.** *Le système (1.1) est coopératif si l'implication suivante est satisfaite :*

$$x \geq x',\ \mathbf{u} \geq \mathbf{u}',\ \mathbf{w} \geq \mathbf{w}' \Rightarrow \forall t \geq 0,\ \Phi(t, x, \mathbf{u}, \mathbf{w}) \geq \Phi(t, x', \mathbf{u}', \mathbf{w}').$$

Cette définition est illustrée dans la Figure 1 pour le cas scalaire ($n = p = q = 1$).

L'extension de la condition de Kamke-Müller aux systèmes avec des variables d'entrée nous permet de caractériser un système coopératif sans avoir à connaître explicitement ses trajectoires.

Proposition 1.6. *Un système (1.1) localement Lipschitz est coopératif si et seulement si on a pour tout $i \in \{1, \ldots, n\}$:*

$$x \geq x',\ x_i = x_i',\ u \geq u',\ w \geq w' \Rightarrow f_i(x, u, w) \geq f_i(x', u', w').$$

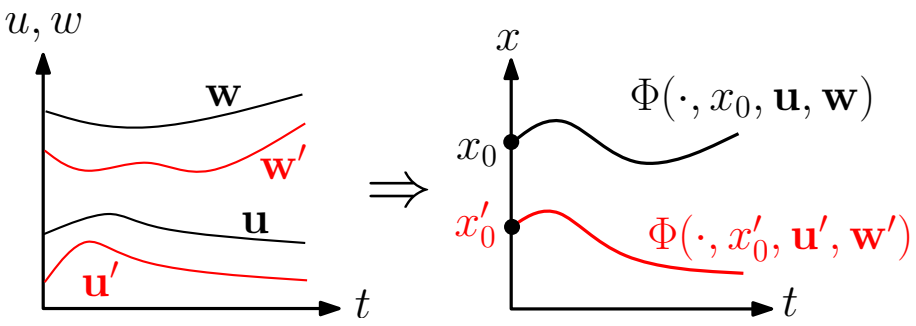


Figure 1 – Exemple scalaire d'un système coopératif.

Pour un système continûment différentiable, cette propriété est vérifiée de manière équivalente en s'intéressant au signe des dérivées partielles du champ de vecteurs f.

Proposition 1.7. *Un système (1.1) continûment différentiable est coopératif si et seulement si on a pour tout* $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$, $w \in \mathbb{R}^q$, $i, j \in \{1, \dots, n\}$, $j \neq i$, $k \in \{1, \dots, p\}$, $l \in \{1, \dots, q\}$ *:*

$$\frac{\partial f_i}{\partial x_j}(x, u, w) \geq 0, \quad \frac{\partial f_i}{\partial u_k}(x, u, w) \geq 0, \quad \frac{\partial f_i}{\partial w_l}(x, u, w) \geq 0.$$

On suppose que les deux variables d'entrée du système (1.1) sont bornées dans des intervalles :

$$u \in [\underline{u}, \overline{u}] \subseteq \mathbb{R}^p \text{ et } w \in [\underline{w}, \overline{w}] \subseteq \mathbb{R}^q.$$

Combiner ces variables d'entrée bornées avec la définition d'un système coopératif joue un rôle important dans la suite de cette thèse pour l'analyse de la robustesse. En effet, toute trajectoire du système (1.1) peut ainsi être encadrée par les deux trajectoires du système utilisant les valeurs extrêmes des entrées de contrôle et de perturbation.

Chapitre 2 : Invariance contrôlée robuste

Pour un système perturbé avec des perturbations inconnues, la notion classique de stabilité n'est généralement pas utilisable puisque l'asservissement du système n'est pas capable d'anticiper les valeurs de perturbations. Il nous faut donc utiliser des variantes de cette notion correspondant au problème de contrôle robuste. En particulier, lorsque les perturbations sont bornées comme c'est le cas dans cette thèse, il est possible de considérer la notion de *stabilité en pratique* (practical stability en anglais) introduite dans [LSL61]. Cette notion étend la définition de la stabilité classique en un point à une stabilité dans un ensemble : les perturbations empêchent la stabilité en un point particulier, mais leurs valeurs bornées nous permettent de contrôler l'état du système pour qu'il reste dans un ensemble autour de ce point.

Plutôt que de relâcher la notion de stabilité comme décrit précédemment, dans ce chapitre nous nous intéressons directement au contrôle du système à l'intérieur d'un ensemble de l'espace d'état, sans spécifier un point particulier autour duquel le système doit se stabiliser. Ce type d'objectif est lié à la notion d'invariance [Bla99] :

un ensemble $\mathcal{S}$ est un invariant si l'état d'un système initialisé dans $\mathcal{S}$ reste à tout instant dans $\mathcal{S}$. Le fait que le système (1.1) soit coopératif simplifie grandement les notions d'invariance et de robustesse pour caractériser des ensembles invariants (contrôlés) robustes.

Invariance robuste

Dans un premier temps, l'entrée de contrôle u est laissée libre et est donc considérée comme une perturbation, au même titre que w. Dans ce cas, la notion d'invariance associée au système (1.1) est l'*invariance robuste*, avec une notion de robustesse par rapport aux deux variables d'entrées u et w.

Définition 2.2 (Invariance robuste)**.** *Un ensemble $\mathcal{S}$ est un invariant robuste pour (1.1) si,*

$$\forall x_0 \in \mathcal{S},\ \forall \mathbf{u} \in [\underline{u}, \overline{u}],\ \forall \mathbf{w} \in [\underline{w}, \overline{w}],\ \forall t \geq 0,\ \Phi(t, x_0, \mathbf{u}, \mathbf{w}) \in \mathcal{S}.$$

Il est clair à partir de cette définition que si l'état est initialisé dans un ensemble invariant robuste $\mathcal{S}$, tous les états atteignables par le système (1.1) sont contenus dans $\mathcal{S}$, mais la réciproque n'est pas vraie. Pour un système coopératif, un ensemble invariant robuste peut être très simplement caractérisé si cet ensemble est un intervalle de l'espace d'état. En particulier, cette caractérisation ne nécessite pas la connaissance des trajectoires Φ du système, mais simplement son champ de vecteurs f et les valeurs extrêmes de ses trois variables. De plus, si pour des fonctions d'entrée constantes $\mathbf{u}(t) = u$ et $\mathbf{w}(t) = w$ (pour tout $t \geq 0$) il existe un unique point d'équilibre $k_x(u, w)$ du système (1.1), alors il est également possible de définir l'intervalle invariant robuste minimal au sens de l'inclusion.

Théorème 2.3. *L'intervalle $[\underline{x}, \overline{x}] \subseteq \mathbb{R}^n$ est un invariant robuste si et seulement si*

$$\begin{cases} f(\overline{x}, \overline{u}, \overline{w}) \leq 0, \\ f(\underline{x}, \underline{u}, \underline{w}) \geq 0. \end{cases}$$

Si on a l'existence et l'unicité des points d'équilibre du système (1.1), alors l'intervalle invariant robuste minimal est $[k_x(\underline{u}, \underline{w}), k_x(\overline{u}, \overline{w})]$.

Cet intervalle invariant robuste minimal est particulièrement utile pour restreindre l'étude du système à cet ensemble lors des implémentations numériques.

Invariance contrôlée robuste

De manière similaire, il est possible de définir la notion d'invariance contrôlée robuste en utilisant un retour d'état $\mathsf{u} : \mathbb{R}^n \to \mathbb{R}^p$ pour forcer l'invariance robuste, où la robustesse ne dépend plus que de l'entrée de perturbation w. Dans la définition qui suit, on note Φ_{u} les trajectoires du système en boucle fermée avec le retour d'état u.

Définition 2.4 (Invariance contrôlée robuste)**.** *Un ensemble $\mathcal{S}$ est un invariant contrôlé robuste si il existe un contrôleur $\mathsf{u} : \mathcal{S} \to [\underline{u}, \overline{u}]$ tel que*

$$\forall x_0 \in \mathcal{S},\ \forall \mathbf{w} \in [\underline{w}, \overline{w}],\ \forall t \geq 0,\ \Phi_{\mathsf{u}}(t, x_0, \mathbf{w}) \in \mathcal{S}.$$

u est alors appelé un contrôleur d'invariance dans $\mathcal{S}$.

Comme pour l'invariance robuste dans le Théorème 2.3, une caractérisation d'un intervalle invariant contrôlé robuste peut être obtenue en n'utilisant que le signe du champ de vecteurs de (1.1) avec les valeurs extrêmes de ses variables, mais où cette fois l'entrée de contrôle u s'oppose aux effets de la perturbation w. Ce résultat nécessite le fait que (1.1) soit coopératif ainsi qu'une propriété supplémentaire de *contrôle local* impliquant que chaque composante u_k de l'entrée de contrôle u n'a une influence directe que sur une unique composante x_i de l'état x dans le champ de vecteurs f de (1.1).

Théorème 2.5. *L'intervalle $[\underline{x}, \overline{x}]$ est un invariant contrôlé robuste si et seulement si*

$$\begin{cases} f(\overline{x}, \underline{u}, \overline{w}) \leq 0, \\ f(\underline{x}, \overline{u}, \underline{w}) \geq 0. \end{cases}$$

Ainsi, si la plus petite valeur de contrôle $\underline{u}$ permet une décroissance de tous les états ($f \geq 0$) lorsque l'état est sur la borne supérieure de l'intervalle $\overline{x}$ avec les perturbations maximales $\overline{w}$, et symétriquement le contrôle maximal permet une croissance de tous les états lorsque l'on est dans les conditions minimales de l'état et la perturbation, alors il existe un contrôleur permettant de garder l'état dans cet intervalle pour toutes valeurs de perturbation. Théorème 2.5 défini donc deux ensembles de l'espace d'état : un où la borne supérieure $\overline{x}$ d'un intervalle invariant contrôlé robuste doit être choisie et un où sa borne inférieure $\underline{x}$ doit être choisie.

Un exemple simple d'un contrôleur d'invariance est le contrôleur décentralisé et affine suivant :

$$\mathsf{u}_i(x) = \underline{u_i} + (\overline{u_i} - \underline{u_i}) \frac{\overline{x_i} - x_i}{\overline{x_i} - \underline{x_i}}. \tag{2.4}$$

L'utilisation des valeurs extrêmes de l'intervalle de contrôle n'est pas nécessaire pour obtenir un contrôleur d'invariance et il suffit d'utiliser des valeurs de contrôle qui préservent les inégalités du Théorème 2.5.

Stabilisabilité locale robuste

Cette notion décrit des états dans lesquels le système peut être stabilisé quelque soit la valeur de la perturbation.

Définition 2.7 (Stabilisabilité locale robuste)**.** *L'état x^* est localement stabilisable de manière robuste si pour tout $\varepsilon > 0$, il existe $\delta > 0$ et $\mathsf{u} : \mathcal{B}(x^*, \varepsilon) \to [\underline{u}, \overline{u}]$ tels que :*

$$\forall x_0 \in \mathcal{B}(x^*, \delta),\ \forall \mathbf{w} \in [\underline{w}, \overline{w}],\ \forall t \geq 0,\ \Phi_{\mathsf{u}}(t, x_0, \mathbf{w}) \in \mathcal{B}(x^*, \varepsilon),$$

où $\mathcal{B}(x^*, r)$ représente une boule de rayon r centrée en x^*.

Dans cette définition, x^* est stabilisable si pour toute boule arbitrairement petite autour de x^* il existe une autre boule d'états initiaux tels que le système peut être contrôlé pour rester dans la première boule quelques soient les perturbations. Cette définition peut facilement être modifiée en remplaçant les boules par des petits intervalles invariants contrôlés robustes. Ce changement permet d'obtenir le résultat suivant où la stabilisabilité locale robuste est assimilée à l'invariance contrôlée robuste dans un intervalle réduit à un unique point ($\underline{x} = \overline{x}$).

Théorème 2.8. *L'état x^* est localement stabilisable de manière robuste si*

$$\begin{cases} f(x^*, \underline{u}, \overline{w}) < 0, \\ f(x^*, \overline{u}, \underline{w}) > 0. \end{cases}$$

Si x^* est localement stabilisable de manière robuste, alors

$$\begin{cases} f(x^*, \underline{u}, \overline{w}) \leq 0, \\ f(x^*, \overline{u}, \underline{w}) \geq 0. \end{cases}$$

Cette notion correspond au cas où les deux ensembles définis par Théorème 2.5 pour le choix des bornes d'un intervalle invariant contrôlé robuste ont une intersection non vide.

Stabilisation robuste dans un ensemble

Après s'être intéressé au problème de synthèse d'un contrôleur pour maintenir l'état du système à l'intérieur d'un intervalle (Définition 2.4 et Théorème 2.5), il est maintenant naturel de chercher un contrôleur permettant d'amener l'état du système dans un intervalle lorsque l'état initial se trouve à l'extérieur.

Définition 2.9 (Contrôleur stabilisant)**.** *Un contrôleur $\mathsf{u} : [\underline{x_0}, \overline{x_0}] \to [\underline{u}, \overline{u}]$ est un contrôleur stabilisant de $[\underline{x_0}, \overline{x_0}]$ vers $[\underline{x_f}, \overline{x_f}] \subseteq [\underline{x_0}, \overline{x_0}]$ si*

$$\forall x_0 \in [\underline{x_0}, \overline{x_0}],\ \forall \mathbf{w} \in [\underline{w}, \overline{w}],\ \exists T \geq 0 \mid \forall t \geq T,\ \ \Phi_{\mathsf{u}}(t, x_0, \mathbf{w}) \in [\underline{x_f}, \overline{x_f}].$$

L'idée générale est d'utiliser une famille d'intervalles invariants contrôlés robustes décroissante selon l'inclusion et qui converge vers un intervalle invariant contrôlé robuste final. Pour cela on définit deux fonctions décrivant l'évolution des bornes inférieures et supérieures de la famille d'intervalles.

Hypothèse 4. *Il existe deux fonctions continûment différentiables*

$$\underline{X},\ \overline{X} : [0, 1] \to \mathbb{R}^n,$$

respectivement strictement décroissante et croissante sur toutes leurs composantes

$$\frac{d\underline{X}}{d\lambda}(\lambda) < 0,\ \ \frac{d\overline{X}}{d\lambda}(\lambda) > 0,\ \ \forall \lambda \in [0, 1],$$

telles que $\underline{X}(0) = \underline{x_f}$, $\underline{X}(1) = \underline{x_0}$, $\overline{X}(0) = \overline{x_f}$, $\overline{X}(1) = \overline{x_0}$ et qui satisfont

$$f(\underline{X}(\lambda), \overline{u}, \underline{w}) > 0,\ \ f(\overline{X}(\lambda), \underline{u}, \overline{w}) < 0,\ \ \forall \lambda \in [0, 1].$$

La dernière partie de cette hypothèse donne $[\underline{x_0}, \overline{x_0}] = [\underline{X}(1), \overline{X}(1)]$, $[\underline{x_f}, \overline{x_f}] = [\underline{X}(0), \overline{X}(0)]$ et pour tout $\lambda,\ \lambda' \in [0, 1]$, $[\underline{X}(\lambda), \overline{X}(\lambda')]$ est un invariant contrôlé robuste. On définit ensuite les fonctions $\underline{\lambda},\ \overline{\lambda} : [\underline{x_0}, \overline{x_0}] \to [0, 1]$

$$\begin{cases} \overline{\lambda}(x) = \min\{\lambda \in [0, 1] \mid \overline{X}(\lambda) \geq x\}, \\ \underline{\lambda}(x) = \min\{\lambda \in [0, 1] \mid \underline{X}(\lambda) \leq x\}. \end{cases} \tag{2.9}$$

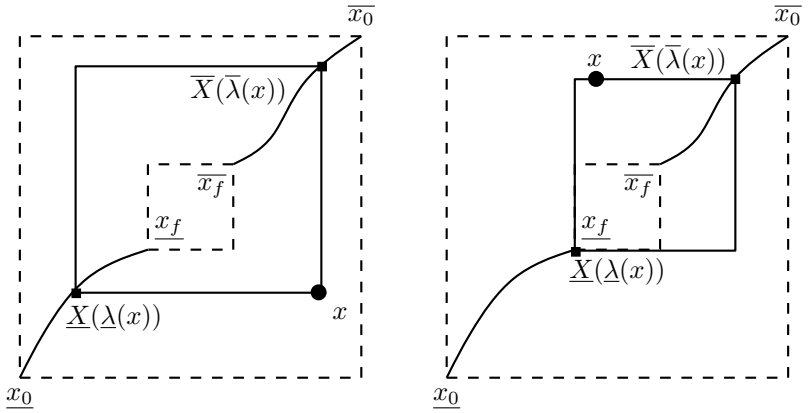


Figure 2 – Plus petit élément $[\underline{X}(\underline{\lambda}(x)), \overline{X}(\overline{\lambda}(x))]$ de la famille paramétrisée d'intervalles invariants contrôlés robustes $[\underline{X}(\lambda), \overline{X}(\lambda')]$ contenant l'état x.

Comme le montre la Figure 2, cela signifie que $[\underline{X}(\underline{\lambda}(x)), \overline{X}(\overline{\lambda}(x))]$ est le plus petit intervalle de la famille paramétrisée $[\underline{X}(\lambda), \overline{X}(\lambda')]$ contenant l'état actuel x. Puisque cet intervalle est un invariant contrôlé robuste avec des inégalités strictes d'après Hypothèse 4, il est possible de trouver un contrôleur d'invariance pour cet intervalle pouvant forcer l'état vers l'intérieur de l'intervalle, garantissant ainsi une stricte décroissance des fonctions $\underline{\lambda}$ et $\overline{\lambda}$ qui agissent comme des fonctions de Lyapunov. En s'inspirant du contrôleur décentralisé (2.4), on peut définir un contrôleur correspondant à notre problème et prouver qu'il est stabilisant.

Théorème 2.11. *Sous l'Hypothèse 4, le contrôleur* u *défini par (2.9) et*

$$\mathsf{u}_i(x) = \underline{u_i} + (\overline{u_i} - \underline{u_i}) \frac{\overline{X}_i(\overline{\lambda}(x)) - x_i}{\overline{X}_i(\overline{\lambda}(x)) - \underline{X}_i(\underline{\lambda}(x))}$$

est un contrôleur stabilisant de $[\underline{x_0}, \overline{x_0}]$ *vers* $[\underline{x_f}, \overline{x_f}]$.

Le contrôleur présenté dans Théorème 2.11 n'est qu'un exemple de contrôleur stabilisant mais il n'est pas le seul qui existe. Par exemple, il est suffisant que la valeur du contrôleur $\mathsf{u}(x)$ permette une stricte décroissance de $\underline{\lambda}$ et $\overline{\lambda}$.

Le résultat du Théorème 2.11 est basé sur l'existence des fonctions $\underline{X}$ et $\overline{X}$ satisfaisant l'Hypothèse 4. Nous présentons ci-dessous trois choix possibles pour obtenir de telles fonctions. Il est a noter que toutes ne satisfont pas nécessairement toutes les conditions de l'Hypothèse 4, qu'il faudra donc vérifier numériquement. La première possibilité est de prendre les droites entre $\overline{x_0}$ et $\overline{x_f}$ et entre $\underline{x_0}$ et $\underline{x_f}$:

$$\begin{cases} \overline{X}(\lambda) = \lambda \overline{x_0} + (1-\lambda)\overline{x_f}, \\ \underline{X}(\lambda) = \lambda \underline{x_0} + (1-\lambda)\underline{x_f}. \end{cases} \tag{2.11}$$

Ces fonctions sont respectivement strictement croissante et décroissante si $\overline{x_0} > \overline{x_f}$ et $\underline{x_0} < \underline{x_f}$ mais il est nécessaire de vérifier que les intervalles $[\underline{X}(\lambda), \overline{X}(\lambda')]$ sont des invariants contrôlés robustes.

Une seconde possibilité est de définir, quand c'est possible, les bornes des intervalles $[\underline{x_0}, \overline{x_0}]$ et $[\underline{x_f}, \overline{x_f}]$ comme des points d'équilibre du système en utilisant la fonction $k_x(u, w)$ introduite pour le Théorème 2.3 sur l'invariance robuste :

$$\begin{cases} \exists \overline{u_f}, \overline{u_0} \in [\underline{u}, \overline{u}] \mid & \underline{u} < \overline{u_f} < \overline{u_0}, \quad \overline{x_0} = k_x(\overline{u_0}, \overline{w}), \quad \overline{x_f} = k_x(\overline{u_f}, \overline{w}), \\ \exists \underline{u_f}, \underline{u_0} \in [\underline{u}, \overline{u}] \mid & \underline{u_0} < \underline{u_f} < \overline{u}, \quad \underline{x_0} = k_x(\underline{u_0}, \underline{w}), \quad \underline{x_f} = k_x(\underline{u_f}, \underline{w}). \end{cases}$$

Dans ce cas, les fonctions $\underline{X}$ et $\overline{X}$ peuvent être définie comme des points d'équilibre en utilisant une combinaison convexe des valeurs de contrôle $\underline{u_0}$, $\underline{u_f}$, $\overline{u_0}$ et $\overline{u_f}$:

$$\begin{cases} \overline{U}(\lambda) = \lambda \overline{u_0} + (1-\lambda)\overline{u_f}, & \overline{X}(\lambda) = k_x(\overline{U}(\lambda), \overline{w}), \\ \underline{U}(\lambda) = \lambda \underline{u_0} + (1-\lambda)\underline{u_f}, & \underline{X}(\lambda) = k_x(\underline{U}(\lambda), \underline{w}). \end{cases} \tag{2.12}$$

Ces fonctions satisfont l'Hypothèse 4 si la matrice jacobienne $\partial f/\partial x$ est inversible et que $\partial f_i/\partial u_i > 0$ pour tout i, où u_i représente le vecteur des composantes de contrôle ayant une influence directe sur l'état x_i.

Pour la troisième proposition, on considère les trajectoires du système entre les bornes des intervalles $[\underline{x_0}, \overline{x_0}]$ et $[\underline{x_f}, \overline{x_f}]$ avec les entrée de contrôles constantes introduites dans le paragraphe précédent. Cette solution existe en deux versions selon le sens des trajectoires :

$$\begin{cases} \overline{X}(\lambda) = \Phi(\dfrac{\lambda}{1-\lambda}, \overline{x_f}, \overline{u_0}, \overline{w}), \\ \underline{X}(\lambda) = \Phi(\dfrac{\lambda}{1-\lambda}, \underline{x_f}, \underline{u_0}, \underline{w}), \end{cases} \tag{2.13}$$

de $\overline{x_f}$ et $\underline{x_f}$ à $\overline{x_0}$ et $\underline{x_0}$ ou, dans la direction opposée $\overline{x_0}$ et $\underline{x_0}$ à $\overline{x_f}$ et $\underline{x_f}$:

$$\begin{cases} \overline{X}(\lambda) = \Phi(\dfrac{1-\lambda}{\lambda}, \overline{x_0}, \overline{u_f}, \overline{w}), \\ \underline{X}(\lambda) = \Phi(\dfrac{1-\lambda}{\lambda}, \underline{x_0}, \underline{u_f}, \underline{w}). \end{cases} \tag{2.14}$$

L'implémentation de ces solutions est plus simple que pour la proposition précédente (2.12) puisqu'elle ne nécessite le calcul que de deux points d'équilibre, alors que (2.12) nécessite une connaissance explicite de la fonction k_x. Cet avantage vient avec l'inconvénient que les deux conditions de l'Hypothèse 4 ne sont pas naturellement satisfaites et doivent donc être vérifiées numériquement.

Chapitre 3 : Contrôle symbolique d'un système coopératif

L'objectif de ce chapitre est de synthétiser un contrôleur pour le système (1.1) à partir d'une abstraction de ce système. L'utilisation d'une abstraction nous permet d'une part de simplifier le travail de synthèse du contrôleur si le système original (1.1) est trop compliqué et d'autre part, d'utiliser des méthodes de synthèse appartenant à la théorie du contrôle des systèmes discrets. Pour que le contrôleur synthétisé pour l'abstraction soit applicable au système original, il est nécessaire d'avoir une relation comportementale entre les deux modèles afin de s'assurer que tout comportement

observé sur le système original a un équivalent dans l'abstraction. L'objectif de contrôle est en deux parties. Dans un premier temps, on s'intéresse à une spécification de sûreté pour garder l'état du système dans un intervalle $[\underline{x}, \overline{x}]$. Dans un second temps, on souhaite la minimisation d'un critère de coût pour choisir les valeurs de contrôle optimales parmi celles permettant la sûreté.

Abstraction symbolique

Puisque le système (1.1) en temps continu ne peut pas être décrit sous la forme d'un système de transitions, on s'intéresse à une version échantillonnée de ce système avec une période d'échantillonnage τ constante. Ce système échantillonné est noté $S = (X, X^0, U, \longrightarrow)$ où $X = \mathbb{R}^n$ est l'ensemble des états, $X^0 = [\underline{x}, \overline{x}] \subseteq \mathbb{R}^n$ l'ensemble des états initiaux, $U = [\underline{u}, \overline{u}] \subseteq \mathbb{R}^p$ l'ensemble des entrées et $\longrightarrow$ la relation de transitions définie par :

$$x \xrightarrow{u} x' \iff \exists \mathbf{w} : [0, \tau] \to [\underline{w}, \overline{w}] \mid x' = \Phi(\tau, x, u, \mathbf{w}).$$

On note $Post(x, u) = \{x' \in X \mid x \xrightarrow{u} x'\}$ l'ensemble des successeurs d'un état x de S avec une entrée u.

L'abstraction de S est dénoté $S_a = (X_a, X_a^0, U_a, \underset{a}{\longrightarrow})$. L'ensemble X_a^0 est une partition uniforme de l'intervalle $[\underline{x}, \overline{x}]$ en un ensemble de plus petits intervalles de taille identique. Un élément de la partition $s \in X_a^0$ est noté de manière équivalente $s = [\underline{s}, \overline{s}] \subseteq \mathbb{R}^n$ où $\underline{s}$ et $\overline{s}$ sont les bornes inférieures et supérieures de l'intervalle s. Les éléments de X_a^0 sont appelé des *symboles* puisqu'ils représentent le comportement de tous les états continus qu'ils contiennent. L'ensemble X_a des états de S_a est obtenu en ajoutant à X_a^0 un unique symbole $Out = \mathbb{R}^n \backslash [\underline{x}, \overline{x}]$ représentant tout l'extérieur de l'intervalle pour que X_a soit une partition de $\mathbb{R}^n$. L'ensemble des entrées U_a provient d'une discrétisation uniforme de $[\underline{u}, \overline{u}]$. Puisque le système (1.1) est coopératif, on peut facilement calculer une sur-approximation de l'ensemble des états continus atteignables en un temps τ à partir des états contenus dans un symbole $s = [\underline{s}, \overline{s}]$ et avec une entrée de contrôle constante $u \in U_a$:

$$\forall x \in s,\ \forall \mathbf{w} : [0, \tau] \to [\underline{w}, \overline{w}],\ \Phi(\tau, x, u, \mathbf{w}) \in [\Phi(\tau, \underline{s}, u, \underline{w}), \Phi(\tau, \overline{s}, u, \overline{w})]. \tag{3.5}$$

Les transitions de l'abstraction symbolique peuvent alors être définies en cherchant les symboles partiellement couverts par cette sur-approximation :

$$\forall s \in X_a^0,\ u \in U_a,\ s' \in X_a,\ s \xrightarrow[a]{u} s' \iff s' \cap [\Phi(\tau, \underline{s}, u, \underline{w}), \Phi(\tau, \overline{s}, u, \overline{w})] \neq \emptyset.$$

Grâce à l'utilisation de cette sur-approximation, il est possible de prouver qu'il existe une relation de simulation alternée entre l'abstraction symbolique S_a et le système échantillonné S. Cette relation signifie que pour tout état $x \in X$, le symbole $s \in X_a$ tel que $x \in s$ et pour tout contrôle $u \in U_a \subseteq U$, les successeurs x' de x dans S appartiennent à un successeur s' de s dans S_a :

$$x \in s,\ u \in U_a,\ x' \in Post(x, u) \Rightarrow \exists s' \in Post_a(s, u) \mid x' \in s'.$$

Cela nous permet d'affirmer qu'un contrôleur synthétisé sur S_a pour satisfaire une spécification sur les symboles va satisfaire la même spécification s'il est appliqué au système original S.

Synthèse de sûreté

L'objectif de sûreté de S est de maintenir son état à l'intérieur de l'intervalle $[\underline{x}, \overline{x}]$. Pour l'abstraction symbolique, cela correspond à assurer la sûreté dans la partition X_a^0 de cet intervalle. Pour résoudre ce problème de sûreté, on va chercher le point-fixe de l'opérateur suivant :

$$F_{X_a^0}(Z) = \{s \in Z \cap X_a^0 \mid \exists u \in U_a,\ Post_a(s,u) \subseteq Z\}. \tag{3.6}$$

Pour un ensemble de symboles $Z \subseteq X_a^0$, cet opérateur retourne l'ensemble des symboles de Z dont les successeurs restent dans Z au moins pour une valeur de contrôle $u \in U_a$. Puisque l'abstraction symbolique S_a est un système de transitions fini, le point-fixe maximal $Z_a = \lim_{k\to\infty} F^k_{X_a^0}(X_a^0)$ est obtenu en un nombre fini d'itérations et ce point-fixe correspond à l'ensemble maximal de sûreté pour S_a. Cet ensemble est associé à un contrôleur de sûreté $C_a : Z_a \to 2^{U_a}$:

$$C_a(s) = \{u \in U_a \mid Post_a(s,u) \subseteq Z_a\}, \tag{3.7}$$

associant à chaque symbole s l'ensemble des contrôles pour lesquels tous les successeurs restent dans Z_a. Grâce à la relation de simulation alternée entre S_a et S, il est possible d'utiliser ce contrôleur pour satisfaire la sûreté de S dans l'intervalle $[\underline{x}, \overline{x}]$.

Théorème 3.7. *Le contrôleur C_a^X défini par $C_a^X(x) = C_a(s)$ si $x \in s$ est un contrôleur de sûreté pour S dans l'ensemble $Z_a^X = \{x \in X \mid \exists s \in Z_a,\ x \in s\}$.*

Optimisation des performances

Puisque le contrôleur de sûreté C_a^X donne l'ensemble des valeurs de contrôle sûres associées à un état, on s'intéresse maintenant à choisir la valeur optimale selon un critère de performance. Pour une trajectoire $(x^0, u^0, x^1, u^1, \dots)$ du système S contrôlé avec C_a^X, on souhaite minimiser le critère

$$\sum_{k=0}^{+\infty} \lambda^k g(x^k, u^k), \tag{3.1}$$

où $g(x,u)$ est le coût lié à l'utilisation du contrôle u à partir de l'état x et $\lambda \in (0,1)$ est un facteur de réduction permettant de réduire l'influence du non-déterminisme dans les étapes futures. Puisque l'on raisonne sur l'abstraction S_a, l'information sur les états x^k n'est pas accessible et il est nécessaire de prendre une sur-approximation de la fonction g en introduisant :

$$g_a(s,u) = \max_{x \in s} g(x,u). \tag{3.10}$$

Enfin, pour permettre de calculer un critère de performance en un nombre fini d'opérations, on considère une approximation de (3.1) par le critère

$$\sum_{k=0}^{N} \lambda^k g_a(s^k, u^k) \tag{3.11}$$

sur un horizon fini de N périodes d'échantillonnage. Cette approximation est valable si N et λ sont choisis tels que la valeur λ^{N+1} en facteur du reste des coûts est négligeable.

La minimisation du critère (3.11) parmi les valeurs de contrôle sûres est alors obtenues grâce à un algorithme de programmation dynamique :

$$J_a^N(s) = \min_{u \in C_a(s)} g_a(s,u), \tag{3.12a}$$

$$J_a^k(s) = \min_{u \in C_a(s)} \left(g_a(s,u) + \lambda \max_{s' \in Post_a(s,u)} J_a^{k+1}(s') \right). \tag{3.12b}$$

On commence à la fin de l'horizon ($k = N$) où on ne minimise que le coût de l'étape actuelle puisque l'on considère que la suite de la trajectoire ($k > N$) a un coût négligeable. Ensuite, pour chaque instant d'échantillonnage précédent, on minimise la somme du coût de l'étape actuelle avec le coût maximum des étapes de l'horizon déjà traitées. La dernière étape nous donne le coût $J_a^0(s)$ correspondant à la minimisation de (3.11) en utilisant les pires prédictions des étapes suivantes. Le résultat de l'algorithme est une suite de valeurs de contrôle $(u^0(s), \dots, u^N(s))$ à utiliser à chaque étape de l'horizon fini si $x^0 \in s$. Puisqu'à chaque instant d'échantillonnage, le symbole actuel est mesuré, on peut considérer une stratégie de contrôle à fenêtre glissante où l'on mesure le symbole s, applique la première valeur $u^0(s)$ de cette suite de contrôle et recommence à la prochaine période d'échantillonnage. Ce contrôleur déterministe est donc obtenu en remplaçant le *min* de la dernière étape ($k = 0$) de (3.12b) par un *argmin* :

$$C_a^*(s) = \arg\min_{u \in C_a(s)} \left(g_a(s,u) + \lambda \max_{s' \in Post_a(s,u)} J_a^1(s') \right). \tag{3.13}$$

La version de C_a^* applicable à l'espace d'état continu prend donc la forme :

$$\forall s \in Z_a,\ \forall x \in s,\ C_a^{*X}(x) = C_a^*(s). \tag{3.14}$$

Enfin on note M_a la pire valeur, parmi les symboles sûrs, de la minimisation du coût g_a sur les contrôles sûrs :

$$M_a = \max_{s \in Z_a} \min_{u \in C_a(s)} g_a(s,u). \tag{3.15}$$

Il est alors possible d'obtenir des garanties de performances sur le critère (3.1) de S alors que le contrôleur C_a^{*X} a été obtenu à partir d'une optimisation sur (3.11) pour un horizon fini de S_a.

Théorème 3.10. *Soit $(x^0, u^0, x^1, u^1, \dots)$ avec $x^0 \in Z_a^X$ une trajectoire de S contrôlé avec C_a^{*X} et $s^0, s^1, \dots$ les symboles tels que $x^k \in s^k$ pour tout $k \in \mathbb{N}$. Alors, pour tout $k \in \mathbb{N}$,*

$$\sum_{j=0}^{+\infty} \lambda^j g(x^{k+j}, u^{k+j}) \leq J_a^0(s^k) + \frac{\lambda^{N+1}}{1-\lambda} M_a.$$

Ce résultat signifie que quelque soit l'état x^k de la trajectoire à partir duquel le critère de performance (3.1) est calculé, il est possible de fournir une borne supérieure à la valeur de ce critère. Cette borne supérieure est constituée de deux parties. $J_a^0(s^k)$ est la minimisation dans le pire cas du critère (3.11) sur un horizon fini de S_a, et on a donc naturellement $J_a^0(s^k) \geq \sum_{j=0}^{N} \lambda^j g(x^{k+j}, u^{k+j})$ grâce à la définition (3.10) de g_a. La partie constante $\frac{\lambda^{N+1}}{1-\lambda} M_a$ de la borne supérieure correspond au reste de la trajectoire : le contrôle à fenêtre glissante nous assure que l'on va au moins minimiser le coût actuel $g_a(s^{k+j}, u^{k+j})$ sur les contrôles sûrs ($u^{k+j} \in C_a(s^{k+j})$), mais on ne sait pas à l'avance dans quel symbole s^{k+j} on sera, donc il faut prendre le pire cas sur $s^{k+j} \in Z_a$, ce qui nous donne $M_a \sum_{j=N+1}^{+\infty} \lambda^j$.

Chapitre 4 : Approche compositionnelle du contrôle symbolique

L'approche symbolique présentée dans le chapitre précédent pour la synthèse de contrôleurs souffre d'un problème de passage à l'échelle. En effet, les étapes de création de l'abstraction symbolique et de synthèse du contrôleur déterministe à partir de l'algorithme de programmation dynamique ont une complexité exponentielle en les dimensions n et p de l'espace d'état et de l'espace de contrôle, respectivement. Ces méthodes ne sont donc envisageables que pour les systèmes de très faible dimension. Pour s'attaquer à ce problème, nous considérons maintenant une approche compositionnelle où les méthodes symboliques précédemment introduites sont appliquées à des descriptions partielles du système global (1.1) avant de recomposer les contrôleurs partiels obtenus.

Une approche compositionnelle classique voudrait que la synthèse des contrôleurs sur chaque sous-système soit réalisée indépendamment de ce qui se passe dans les autres sous-systèmes. Ce type de considérations est généralement trop restrictif car la possibilité de synthétiser un contrôleur sur le système global vient des interconnexions entre les différents éléments du système, mais les sous-systèmes pris séparément peuvent ne pas être contrôlables. Une seconde approche moins restrictive, appelée *assume-guarantee*, consiste à synthétiser les contrôleurs d'un sous-système sous certaines contraintes de son environnement correspondant aux hypothèses de bon fonctionnement des autres sous-systèmes. C'est cette approche que nous prendrons lors de la création des abstractions symboliques de chaque sous-système.

Sous-systèmes

On considère que le système original est décomposé en $m \in \mathbb{N}$ sous-systèmes. Chaque sous-système, défini comme une description partielle des dynamiques globales (1.1), peut être caractérisé par 6 ensembles d'index : 4 pour l'état et 2 pour l'entrée de contrôle. Pour l'état, le choix des 4 ensembles est décrit par la Figure 3. On commence par prendre une partition $(I_1^c, \ldots, I_m^c)$ de l'ensemble des index $\{1, \ldots, n\}$, ensuite l'état de chaque sous-système $i \in \{1, \ldots, m\}$ est décrit par les 4 ensembles I_i, I_i^c, I_i^o et K_i :

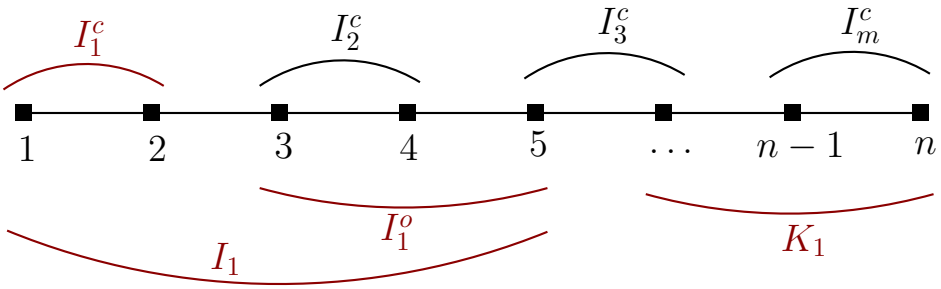


Figure 3 – Partition de $\{1, \ldots, n\}$ et ensembles d'index pour l'état d'un sous-système.

- $I_i \supseteq I_i^c$ représente l'ensemble des états modélisés dans le sous-système ;
- I_i^c est l'ensemble des états que l'on souhaite contrôler ;
- $I_i^o = I_i \backslash I_i^c$ est l'ensemble des états qui sont seulement observés mais pas contrôlés ;
- $K_i = \{1, \ldots, n\} \backslash I_i$ est l'ensemble des états restants, qui ne sont pas observés et qui doivent donc être considérés comme des perturbations.

Le choix des composantes d'entrée associées à un sous-système est similaire mais seulement décrit par 2 ensembles d'index puisque l'on considère que toutes les entrées modélisées sont utilisées pour le contrôle. Nous prenons donc une partition $(J_1, \ldots, J_m)$ de l'ensemble des index de contrôle $\{1, \ldots, p\}$. Les entrées du sous-système $i \in \{1, \ldots, m\}$ sont décrites par :

- J_i, l'ensemble des entrées utilisées pour contrôler les états $x_{I_i^c}$;
- $L_i = \{1, \ldots, p\} \backslash J_i$, les composantes d'entrée restantes, considérées comme des perturbations.

Le rôle de ces 6 ensemble d'index peut être résumé comme suit : pour le sous-système $i \in \{1, \ldots, m\}$, on modélise les états $x_{I_i} = (x_{I_i^c}, x_{I_i^o})$ où $x_{I_i^c}$ sont contrôlés à l'aide des entrées u_{J_i} et $x_{I_i^o}$ sont seulement observés pour augmenter la précision du sous-système, alors que x_{K_i} et u_{L_i} sont considérés comme des perturbations extérieures.

Dans les cas où la notation x_I n'est pas assez claire pour indiquer la projection d'une variable ou d'un ensemble x sur l'espace de dimensions réduites aux index dans I, on utilisera l'opérateur de projection $\pi_I(x) = x_I$.

Pour pouvoir utiliser la méthode d'abstraction symbolique du chapitre précédent sur chaque sous-système, il est nécessaire que toutes les variables considérées comme des perturbations soient bornées. La perturbation classique w ainsi que les variables de contrôle u_{L_i} sont déjà supposées être bornées. Il ne nous reste donc qu'à introduire une obligation d'*assume-guarantee* sur les états non observés, pour lesquels on suppose que les spécifications de sûreté sont satisfaites grâce à l'action des autres sous-systèmes.

Obligation d'Assume-Guarantee 1. *Pour tout* $i \in \{1, \ldots, m\}$, $x_{K_i} \in \pi_{K_i}([\underline{x}, \overline{x}])$.

Une seconde obligation d'*assume-guarantee* est nécessaire pour prendre en compte le fait que seulement une partie des états x_{I_i} modélisés dans le sous-système i doivent être contrôlés ($x_{I_i^c}$), alors que les autres ($x_{I_i^o}$) sont simplement observés pour améliorer la précision du modèle. Nous supposons donc que les spécifications de sûreté pour ces états non-contrôlés sont satisfaites grâce à l'action des autres sous-systèmes.

Obligation d'Assume-Guarantee 2. *Pour tout* $i \in \{1, \ldots, m\}$, $x_{I_i^o} \in \pi_{I_i^o}([\underline{x}, \overline{x}])$.

L'abstraction symbolique $S_i = (X_i, X_i^0, U_i, \underset{i}{\longrightarrow})$ du sous-système i est décrite par les trois ensembles suivants :

- $X_i^0 = \pi_{I_i}(X_a^0)$ est une partition uniforme de $\pi_{I_i}([\underline{x}, \overline{x}])$ en intervalles ;
- $X_i = X_i^0 \cup \{Out_i\}$ est une partition de $\pi_{I_i}(\mathbb{R}^n)$;
- $U_i = \pi_{J_i}(U_a)$ est une discrétisation uniforme de $\pi_{J_i}([\underline{u}, \overline{u}])$.

Pour définir les transitions de S_i, on dénote $RS_i(s_i, u_i)$ la sur-approximation de l'ensemble atteignable de (1.1) en temps τ à partir du symbole $s_i = [\underline{s_i}, \overline{s_i}] \in X_i^0$, avec l'entrée $u_i \in U_i$ et sous l'Obligation d'Assume-Guarantee 1 :

$$RS_i(s_i, u_i) = [\Phi(\tau, (\underline{s_i}, \underline{x}_{K_i}), (u_i, \underline{u}_{L_i}), \underline{w}), \Phi(\tau, (\overline{s_i}, \overline{x}_{K_i}), (u_i, \overline{u}_{L_i}), \overline{w})]. \tag{4.2}$$

Alors, les transitions de S_i sont définies par :

- $\forall s_i \in X_i^0,\ u_i \in U_i,\ s_i' \in X_i^0,\ s_i \underset{i}{\xrightarrow{u_i}} s_i' \iff s_i' \cap \pi_{I_i}(RS_i(s_i, u_i)) \neq \emptyset$;
- $\forall s_i \in X_i^0,\ u_i \in U_i,\ s_i \underset{i}{\xrightarrow{u_i}} Out_i \iff \pi_{I_i^c}(RS_i(s_i, u_i)) \not\subseteq \pi_{I_i^c}([\underline{x}, \overline{x}])$ ou $\pi_{I_i}(RS_i(s_i, u_i)) \cap \pi_{I_i}([\underline{x}, \overline{x}]) = \emptyset$.

Le premier point de cette définition ($s_i, s_i' \in X_i^0$) correspond à la méthode classique où une transition vers le symbole s_i' existe si son intersection avec la sur-approximation de l'ensemble atteignable est non-vide. Pour le second point, ($s_i' = Out_i$) on combine la méthode précédente avec l'Obligation d'Assume-Guarantee 2, ce qui retire certaines transitions vers Out_i :

- si l'ensemble atteignable $RS_i(s_i, u_i)$ sort de l'intervalle $[\underline{x}, \overline{x}]$ sur les dimensions des états contrôlés (I_i^c), $s_i \underset{i}{\xrightarrow{u_i}} Out_i$ existe ;
- si l'ensemble atteignable est entièrement en dehors de l'intervalle, l'Obligation d'Assume-Guarantee 2 ne peut pas être satisfaite et on garde la transition $s_i \underset{i}{\xrightarrow{u_i}} Out_i$;
- dans le reste des cas, c'est à dire quand la possible transition vers Out_i n'est due qu'à une sortie de l'intervalle sur les dimensions des états non-contrôlés (I_i^o), l'Obligation d'Assume-Guarantee 2 empêche cette transition d'exister.

La synthèse des contrôleurs pour la sûreté et les performances de chaque sous-système est ensuite réalisée de manière identique au chapitre précédent. Pour la sûreté du sous-système i, on obtient ainsi un ensemble sûr $Z_i \subseteq X_i^0$ et un contrôleur de sûreté $C_i : Z_i \to 2^{U_i}$ tel que

$$C_i(s_i) = \{u_i \in U_i \mid \emptyset \neq Post_i(s_i, u_i) \subseteq Z_i\}. \tag{4.4}$$

Pour l'optimisation des performances, on utilise une fonction de coût $g_i(s_i, u_i)$ dont la dépendance en s_i n'est liée qu'aux composantes contrôlées de l'état. Le critère de performance $\sum_{k=0}^{N} \lambda^k g_i(s_i^k, u_i^k)$ est minimisé dans le pire cas à l'aide d'un algorithme de programmation dynamique et une stratégie de contrôle à fenêtre glissante est ensuite utilisée pour obtenir le contrôleur déterministe $C_i^* : Z_i \to U_i$:

$$C_i^*(s_i) = \underset{u_i \in C_i(s_i)}{\arg\min} \left(g_i(s_i, u_i) + \lambda \max_{s_i' \in Post_i(s_i, u_i)} J_i^1(s_i') \right). \tag{4.8}$$

Composition

La composition des sous-systèmes donne un système de transitions représentant le modèle global, bien qu'obtenu par des méthodes plus conservatives que celles utilisées pour S_a. Ce système recomposé est décrit par $S_c = (X_c, X_c^0, U_c, \underset{c}{\longrightarrow})$, où $X_c = X_a$, $X_c^0 = X_a^0$, $U_c = U_a$ et les transitions sont définies par :

- $\forall s \in X_c^0,\ u \in U_c,\ s' \in X_c^0,\ s \xrightarrow[c]{u} s' \iff \forall i \in \{1, \ldots, m\},\ s_{I_i} \xrightarrow[i]{u_{J_i}} s'_{I_i}$
- $\forall s \in X_c^0,\ u \in U_c,\ s \xrightarrow[c]{u} Out \iff \exists i \in \{1, \ldots, m\} \mid s_{I_i} \xrightarrow[i]{u_{J_i}} Out_i$

La première condition signifie qu'une transition entre deux symboles de X_c^0 existe dans S_c si sa projection existe dans chaque sous-système. Dans la seconde condition, il suffit qu'un sous-système ait une transition vers l'extérieur de l'intervalle pour qu'une telle transition existe aussi dans S_c. Comme pour l'abstraction symbolique globale S_a du précédent chapitre, il est possible de prouver qu'il existe une relation de simulation alternée entre S_c et le système original S, ce qui signifie que l'on a l'implication suivante :

$$x \in s,\ u \in U_c,\ x' \in Post(x, u) \Rightarrow \exists s' \in Post_c(s, u) \mid x' \in s'.$$

Grâce à cette relation de simulation alternée, il est alors possible de montrer que la composition des contrôleurs de sûreté C_i est aussi un contrôleur de sûreté pour S.

Théorème 4.3. *Le contrôleur $C_c^X : X \to 2^U$ défini par $C_c^X(x) = C_1(s_{I_1}) \times \cdots \times C_m(s_{I_m})$ si $x \in s$ est un contrôleur de sûreté pour S dans $Z_c^X = \{x \in X \mid \exists s \in X_c^0,\ \forall i \in \{1, \ldots, m\},\ \pi_{I_i}(x) \in s_{I_i}$ and $s_{I_i} \in Z_i\}$.*

Du fait des plus larges sur-approximations des ensembles atteignables (4.2) dans cette méthode compositionnelle par rapport à la méthode centralisée du chapitre précédent, il est naturel que l'on obtienne des résultats plus faibles en terme de sûreté.

Corollaire 4.4. $Z_c^X \subseteq Z_a^X$.

De la même manière que pour M_a défini dans (3.15), on définit

$$M_i = \max_{s_i \in Z_i} \min_{u_i \in C_i(s_i)} g_i(s_i, u_i), \tag{4.17}$$

et on fait l'hypothèse suivante.

Hypothèse 6. $\forall s \in X_a^0,\ u \in U_a,\ g_a(s,u) \leq \sum_{i=1}^{m} g_i(s_{I_i}, u_{J_i})$. $M_a \leq \sum_{i=1}^{m} M_i$.

Il est alors possible d'obtenir des garanties de performances de forme similaire au Théorème 3.10 en utilisant la composition des contrôleurs C_i^*.

Théorème 4.5. *Soit $(x^0, u^0, x^1, u^1, \dots)$ avec $x^0 \in Z_c^X$ une trajectoire de S contrôlé avec C_c^{*X} défini par $C_c^{*X}(x) = (C_1^*(s_{I_1}), \dots, C_m^*(s_{I_m}))$ si $x \in s$. Soient $s^0, s^1, \dots$ les symboles tels que $x^k \in s^k$ pour tout $k \in \mathbb{N}$. Alors, sous l'Hypothèse 6, pour tout $k \in \mathbb{N}$,*

$$\sum_{j=0}^{+\infty} \lambda^j g(x^{k+j}, u^{k+j}) \leq \sum_{i=1}^{m} J_i^0(s_{I_i}^k) + \frac{\lambda^{N+1}}{1-\lambda} \sum_{i=1}^{m} M_i.$$

Comme pour la sûreté, on obtient aussi des garanties de performances plus faibles avec la méthode compositionnelle.

Corollaire 4.9. *Sous l'Hypothèse 6, on a pour tout $s \in Z_c = \{s \in X_c^0 \mid \forall i \in \{1, \dots, m\},\ s_{I_i} \in Z_i\}$:*

$$J_a^0(s) + \frac{\lambda^{N+1}}{1-\lambda} M_a \leq \sum_{i=1}^{m} J_i^0(s_{I_i}) + \frac{\lambda^{N+1}}{1-\lambda} \sum_{i=1}^{m} M_i.$$

Complexité

Malgré les résultats naturellement plus faibles en termes de sûreté et de performances, l'approche compositionnelle permet une forte réduction de la complexité des étapes d'abstraction du modèle et de synthèse des contrôleurs. Notons $\alpha_x \in \mathbb{N}$ et $\alpha_u \in \mathbb{N}$ les précisions par dimension de la partition de l'intervalle d'état $[\underline{x}, \overline{x}] \subseteq \mathbb{R}^n$ en plus petits intervalles identiques et de la discrétisation de l'intervalle de contrôle $[\underline{u}, \overline{u}] \subseteq \mathbb{R}^p$, respectivement. Cela signifie que l'intervalle $[\underline{x}, \overline{x}]$ est partitionné en α_x^n symboles et que $[\underline{u}, \overline{u}]$ est discrétisé en α_u^p valeurs de contrôle.

Les deux étapes les plus coûteuses en temps de calcul sont la création de l'abstraction symbolique et l'algorithme de programmation dynamique. Pour l'approche symbolique centralisée du Chapitre 3, l'abstraction symbolique est obtenue en calculant 2 successeurs du système échantillonné S pour chaque couple symbole-entrée, ce qui nécessite donc le calcul de $2\alpha_x^n \alpha_u^p$ successeurs de S. Pour la programmation dynamique, à chaque étape de l'horizon fini de N périodes d'échantillonnage, pour chaque couple symbole-entrée il faut étudier le coût de l'ensemble des successeurs, ce qui donne jusqu'à $N\alpha_x^{2n}\alpha_u^p$ itérations. On peut donc voir que la complexité de ces étapes est exponentielle en les dimensions n et p des espaces d'état et de contrôle, polynomiale en les précisions α_x et α_u et linéaire en N.

Dans le cas de l'approche compositionnelle, les complexités ont une forme similaire mais en remplaçant les dimensions n et p par les dimensions des espaces d'état et de contrôle de chaque sous-systèmes. Si l'on note $|I|$ le nombre d'éléments d'un ensemble fini I, les m abstractions symboliques sont obtenues après le calcul de $\sum_{i=1}^{m} 2\alpha_x^{|I_i|}\alpha_u^{|J_i|}$ successeurs du système S et les programmations dynamiques nécessitent un maximum de $\sum_{i=1}^{m} N\alpha_x^{2|I_i|}\alpha_u^{|J_i|}$ itérations. Il faut noter que ces complexités dépendent du nombre d'états *modélisés* (index I_i) et pas seulement des états *contrôlés* (index I_i^c).

Ainsi, à α_x, α_u et N fixés, l'approche compositionnelle peut réduire la complexité de deux manières :

- soit en augmentant le nombre de sous-système, ce qui va nécessairement diminuer le nombre d'éléments dans les ensembles J_i ;
- soit en diminuant la précision des modèles en réduisant le nombre d'états observés mais non-contrôlés (index I_i^o), ce qui va naturellement diminuer le nombre d'éléments dans I_i.

Dans le cas extrême où l'on prend autant de sous-systèmes que de variables d'états ($m = n$) sans autres états observés ($I_i^o = \emptyset$), la complexité devient linéaire en α_x et n.

Chapitre 5 : Contrôle d'un bâtiment intelligent

L'application présentée dans ce chapitre est la motivation principale du travail présenté dans cette thèse et du type de systèmes considérés. Les hypothèses et résultats présentés dans les chapitres précédents sont donc des généralisations de résultats préliminaires initialement obtenus sur cette application.

La consommation d'énergie dans les bâtiments représente jusqu'à 40% de la consommation totale dans les pays développés et cette statistique est en rapide augmentation du fait de la forte croissance de la population mondiale et des demandes en termes de confort. Le concept de bâtiment intelligent (ou bâtiment vert) est apparu dans les années 80 avec les premiers ajouts de solutions technologiques de mesure et de coordination au niveau global du bâtiment permettant de faire des économies d'énergie.

Dans cette thèse, on s'intéresse plus particulièrement à des systèmes de chauffage, ventilation et climatisation (HVAC en anglais). Traditionnellement, dans les bâtiments intelligents, ces actions sont réalisées dans une zone au dessus d'un faux plafond appelée *plenum*. Cela signifie entre autre qu'à la fois l'arrivée et la sortie d'air se trouvent au niveau du plafond, ce qui peut créer des turbulences du fait du mélange forcé entre l'air chaud et l'air froid et donc réduire le confort pour les utilisateurs. Une solution alternative nommée *UnderFloor Air Distribution* (UFAD) permet de résoudre efficacement ces problèmes en plaçant l'arrivée d'air dans un autre plenum situé sous un faux plancher et en conservant la sortie d'air dans le

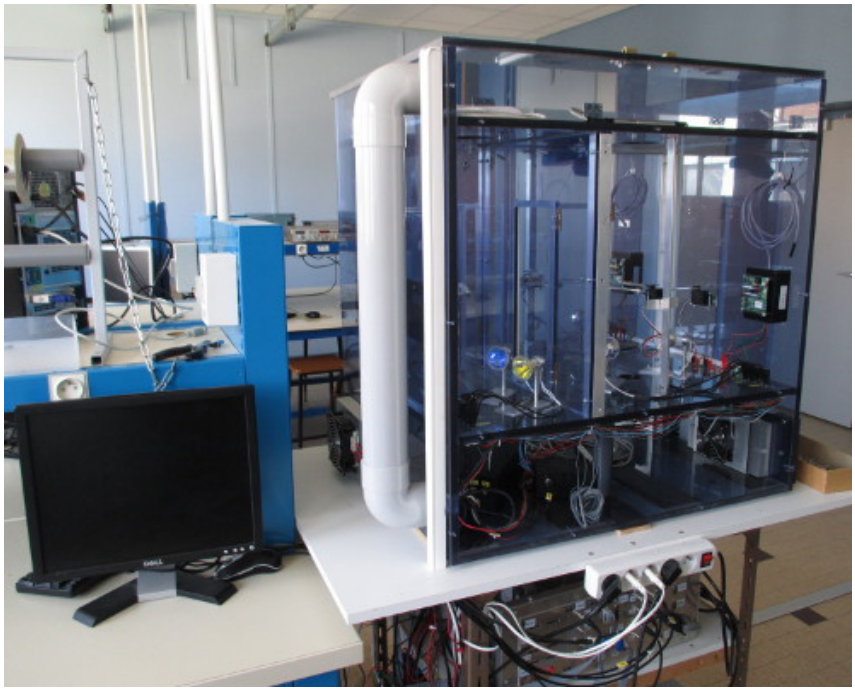


Figure 4 – Maquette d'un bâtiment intelligent de 4 pièces avec la solution UFAD.

plenum du plafond [BD03]. Cela permet un mélange de l'air plus doux où l'arrivée d'air frais au niveau du sol va pousser l'air chaud dans le plenum du plafond grâce à la stratification naturelle des températures.

Description du système

On s'intéresse donc au contrôle de température dans la maquette expérimentale d'un bâtiment équipé de la solution UnderFloor Air Distribution en photo sur la Figure 4. Outre le plenum du sous-sol et du plafond, ce bâtiment est constitué de 4 pièces. Dans chacune de ces pièces, on a un capteur mesurant la température centrale de la pièce et un ventilateur au niveau du plancher envoyant l'air froid du sous-sol dans la pièce. Pour créer des perturbations, on a aussi des lampes halogènes dans chaque pièce pour créer des sources de chaleur et des portes que l'on peut ouvrir entre les pièces.

Dans ce problème de contrôle, on considère que la température du sous-sol est contrôlée séparément et on se concentre donc sur la régulation des températures dans chacune des pièces en jouant sur les actions de ventilation. On se contente donc de modéliser les variations des températures dans les 4 pièces du bâtiment. Pour cela on suppose que la vitesse et la masse de l'air sont suffisamment faibles pour pouvoir négliger son énergie cinétique, son énergie potentielle et pour considérer l'air comme incompressible. On suppose ensuite que la température de chaque pièce est uniforme et que sa valeur est celle mesurée par le capteur : cette hypothèse est similaire à celle d'un *lumped model* où les variations spatiales sont négligées pour obtenir un système à dimensions finies (équations différentielles ordinaires) au lieu d'équations

aux dérivées partielles. Enfin, on suppose que l'air suit la loi des gaz parfaits. Le modèle des variations de température est alors obtenu en combinant les équations de conservations de la masse et de l'énergie dans chaque pièce.

La conservation de la masse dans la pièce i donne, sous l'hypothèse d'incompressibilité de l'air :

$$\dot{m}_{u\to i} - \dot{m}_{i\to c} + \sum_{j\in\mathcal{N}_i} \delta_{d_{ij}} \operatorname{sign}(T_j - T_i)\dot{m}_{d_{ij}} = 0, \tag{5.4}$$

où $\dot{m}_{u\to i}$, $\dot{m}_{i\to c}$ et $\dot{m}_{d_{ij}}$ sont respectivement les débits massiques d'air du sous-sol (index u pour *underfloor*) à la pièce i, de la pièce i au plenum du plafond (index c pour *ceiling*) et entre les pièces i et j lorsque la porte correspondante est ouverte ($\delta_{d_{ij}} = 1$). Les trois débits massiques sont positifs et associés à un signe positif lorsqu'ils entrent dans la pièce i et un signe négatif lorsqu'ils en sortent. En particulier, l'utilisation de la loi des gaz parfaits nous donne que $\dot{m}_{d_{ij}}$ est toujours dirigé de la pièce chaude à la pièce froide, d'où l'utilisation de $\operatorname{sign}(T_j - T_i)$ où T_i et T_j représentent les températures des deux pièces. Enfin, $\mathcal{N}_i$ représente l'ensemble des index des pièces voisines à la pièce i.

La conservation d'énergie dans la pièce i est donnée par :

$$\begin{aligned}\rho V_i C_v \frac{dT_i}{dt} = &\sum_{j\in\mathcal{N}_i^*} \frac{k_{ij}A_{ij}}{\Delta_{ij}}(T_j - T_i) + \delta_{s_i}\varepsilon_{s_i}\sigma A_{s_i}(T_{s_i}^4 - T_i^4) \\ &+ C_p T_u \dot{m}_{u\to i} - C_p T_i \dot{m}_{i\to c} \\ &+ \sum_{j\in\mathcal{N}_i} C_p \max(T_i, T_j)\delta_{d_{ij}} \operatorname{sign}(T_j - T_i)\dot{m}_{d_{ij}}.\end{aligned} \tag{5.8}$$

Dans la partie de gauche de cette équation, on trouve la dérivée de l'énergie dans la pièce, réduite à la seule énergie interne $\rho V_i C_v T_i$ après avoir négligé les énergies cinétique et potentielle de l'air. ρ, V_i et C_v sont la densité de l'air, le volume de la pièce et la capacité thermique de l'air à volume constant. Dans la partie de droite, on trouve trois types de transfert de chaleur.

- La conduction thermique $\frac{k_{ij}A_{ij}}{\Delta_{ij}}(T_j - T_i)$ entre les pièces i et j séparées par un mur de surface A_{ij}, d'épaisseur Δ_{ij} et de conductivité k_{ij}. La zone voisine $j \in \mathcal{N}_i^*$ peut être une autre pièce ($j \in \mathcal{N}_i$), le plenum du sous-sol ($j = u$), le plenum du plafond ($j = c$) ou l'extérieur du bâtiment ($j = o$ pour *outside*).

- La radiation thermique $\delta_{s_i}\varepsilon_{s_i}\sigma A_{s_i}(T_{s_i}^4 - T_i^4)$ provenant des sources de chaleur de température T_{s_i}, de surface A_{s_i}, d'émissivité ε_{s_i}. Ce terme n'apparaît que lorsque la source de chaleur est active : $\delta_{s_i} = 1$.

- Les trois autres transferts de chaleur, de la forme $C_p T_j \dot{m}_{j\to k}$, sont induits par le débit massique d'air de la zone j à la zone k et sont associés à la température T_j de la zone de départ du flux d'air. C_p est la capacité thermique de l'air à pression constante.

Le débit massique $\dot{m}_{u\to i}$ est considéré comme notre entrée de contrôle et celui au niveau d'une porte est calculé à partir du principe de Bernoulli pour les gaz incompressibles : $\dot{m}_{d_{ij}} = \rho A_d \sqrt{2R|T_i - T_j|}$, où A_d est la surface de la porte ouverte et R

est la constante spécifique de l'air dans l'équation des gaz parfaits. En remplaçant dans (5.8) le dernier terme inconnu $\dot{m}_{i\to c}$ par son expression dans (5.4), on obtient le modèle final :

$$\rho V_i C_v \frac{dT_i}{dt} = \sum_{j\in\mathcal{N}_i^*} \frac{k_{ij}A_{ij}}{\Delta_{ij}}(T_j - T_i) + \delta_{s_i}\varepsilon_{s_i}\sigma A_{s_i}(T_{s_i}^4 - T_i^4) \tag{5.9}$$
$$+ C_p\dot{m}_{u\to i}(T_u - T_i) + \sum_{j\in\mathcal{N}_i} \delta_{d_{ij}} C_p \rho A_d \sqrt{2R}\max(0, T_j - T_i)^{3/2}.$$

Le modèle du système global est alors décrit de manière similaire à (1.1) :

$$\dot{T} = f(T, u, w, \delta), \tag{5.12}$$

où $T \in \mathbb{R}^4$ est l'état, $u \in \mathbb{R}^4$ est l'entrée de contrôle avec $u_i = -\dot{m}_{u\to i}$, $w = [T_u, T_c, T_o] \in \mathbb{R}^3$ est une entrée de perturbation regroupant toutes les températures non contrôlées et $\delta \in \{0,1\}^8$ est une entrée de perturbation discrète avec l'état binaire des sources de chaleur et des portes. En calculant les dérivées partielles de f comme dans la Proposition 1.7, on peut prouver que (5.12) est un système coopératif.

Aux vues des nombreuses hypothèses et simplifications faites pour obtenir ce modèle physique des variations de température, nous choisissons de réaliser une identification de type *gray-box* où l'on impose la forme du modèle (5.9) en se délestant du sens physique des paramètres. Pour cela, on regroupe tous les paramètres de (5.9) en un nombre réduit d'inconnues à identifier :

$$\frac{dT_i}{dt} = \sum_{j\in\mathcal{N}_i^*} a_{ij}(T_j - T_i) + \delta_{s_i} b_i (T_{s_i}^4 - T_i^4) \tag{5.11}$$
$$+ c_i u_i (T_u - T_i) + \sum_{j\in\mathcal{N}_i} \delta_{d_{ij}} d_{ij} \max(0, T_j - T_i)^{3/2}.$$

On obtient un problème d'identification à 40 paramètres inconnus (10 par pièce) que l'on résout à l'aide d'un algorithme des moindres carrés à partir de 16 heures de données expérimentales couvrant les principaux transferts de chaleur modélisés dans (5.9). Le modèle (5.11) associé aux valeurs des paramètres identifiés est ensuite évalué sur un scénario expérimental non inclus dans les données utilisées pour l'identification.

Plusieurs limitations sont observées sur ce modèle identifié. La première est liée au fait que les flux d'air à travers les portes ouvertes ne sont en réalité pas unidirectionnels, ce qui crée des transferts de chaleur non modélisés dans la pièce la plus chaude. Le modèle (5.9) ne fait apparaître que la conduction thermique au niveau des murs alors qu'une modélisation plus précise ferait également intervenir la convection entre l'air et les murs. Le principal problème de cet ajout et que le coefficient de convection dépend fortement de la ventilation utilisée dans la pièce, ce qui rend le modèle nettement plus compliqué et lui fait aussi perdre le fait d'être coopératif. Enfin, le terme de radiation thermique est supposé apparaître et disparaître immédiatement après un changement de l'état binaire δ_{s_i} de la lampe, alors qu'en réalité la température finale de la lampe n'est atteinte qu'après un délai. Malgré ces limitations et possibles améliorations du modèle, nous conservons ce modèle (5.11) associé aux paramètres identifiés qui décrit le système réel de manière satisfaisante.

Evaluation des contrôleurs

Ce modèle peut alors être utilisé pour synthétiser les différentes stratégies de contrôle développées dans les chapitres précédents.

Invariance et stabilisation On implémente d'abord les méthodes d'invariance contrôlée robuste et de stabilisation robuste dans un ensemble. Pour cela, on se place dans les conditions où la température du sous-sol est contrôlée à 17 °C grâce à un PID et les autres températures (plenum du plafond et extérieur) varient dans l'intervalle $[22, 25]$. Ces conditions nous permettent de caractériser les deux ensembles de l'espace d'état dans lesquels les bornes d'un intervalle doivent être choisies pour satisfaire l'invariance contrôlée robuste. On choisit alors l'intervalle invariant contrôlé robuste suivant :

$$\underline{T_f} = \begin{pmatrix} 21 & 21 & 21 & 21 \end{pmatrix}, \qquad \overline{T_f} = \begin{pmatrix} 24 & 26 & 26 & 27 \end{pmatrix}.$$

Pour implémenter la stabilisation robuste dans cet intervalle, on opte pour les fonctions affines suivantes :

$$\begin{cases} \overline{X}(\lambda) = \lambda \overline{T_0} + (1-\lambda)\overline{T_f}, \\ \underline{X}(\lambda) = \lambda \underline{T_0} + (1-\lambda)\underline{T_f}, \end{cases}$$

entre les bornes $\underline{T_f}$ et $\overline{T_f}$ de notre intervalle cible et les bornes $\underline{T_0}$ et $\overline{T_0}$ de l'intervalle invariant robuste minimal calculé grâce au Théorème 2.3. Enfin, on utilise le contrôleur stabilisant donné en exemple dans le Théorème 2.11 :

$$\mathsf{u}_i(T) = \underline{u_i} + (\overline{u_i} - \underline{u_i}) \frac{\overline{X}_i(\overline{\lambda}(T)) - T_i}{\overline{X}_i(\overline{\lambda}(T)) - \underline{X}_i(\underline{\lambda}(T))}.$$

Les mesures des températures du bâtiment expérimental ainsi contrôlé sont donnée dans la Figure 5.7 page 136. Outre le fait que la stabilisation et l'invariance robuste sont correctement réalisées, on observe lors de la phase de stabilisation que l'on a toujours une des température qui limite la décroissance des bornes de la famille d'intervalles stabilisants et que dans la pièce correspondante, l'entrée de contrôle prend nécessairement une de ses valeurs extrémales.

Contrôle symbolique centralisé D'une manière similaire, un intervalle invariant contrôlé robuste est choisi dans les conditions où $T_c, T_o \in [21, 24]$. On décide de créer une partition de cet intervalle en plus petits intervalles identiques, en prenant $\alpha_x = 10$ intervalles par dimension, ce qui fait une partition contenant 10000 symboles. L'intervalle de contrôle est discrétisé en $\alpha_u = 4$ valeurs par dimension, ce qui nous donne 256 contrôles discrets possibles. En prenant en compte la précision de la partition, on choisit une période d'échantillonnage $\tau = 34\,\mathrm{s}$ permettant de s'assurer que dans les conditions des dynamiques les plus rapides, la sur-approximation de l'ensemble atteignable atteint des symboles au-delà des voisins immédiats du symbole de départ. La fonction de coût pour l'abstraction symbolique prend un compromis entre la minimisation de trois critères :

$$g_a(s^k, u^k, u^{k-1}) = \frac{\|u^k\|}{\|\overline{u} - \underline{u}\|} + \frac{\|u^k - u^{k-1}\|}{\|\overline{u} - \underline{u}\|} + \frac{\|s_*^k - T_*\|}{\|(\overline{T} + \underline{T})/2\|}. \tag{5.14}$$

On minimise donc la norme de l'entrée de contrôle, ses variations et la distance entre le centre s_*^k du symbole actuel s^k et le centre T_* de l'intervalle $[\underline{T}, \overline{T}]$. Ces trois critères sont normalisés par rapport à leurs valeurs maximales pour leur donner une influence équivalente. Enfin, l'algorithme de programmation dynamique est réalisé sur un horizon de $N = 5$ périodes d'échantillonnage avec un facteur de réduction $\lambda = 0.5$. Ces choix permettent de s'assurer que le coût des étapes suivantes (sur un horizon infini) ont une influence négligeable : $\lambda^{N+1} \approx 1.6\%$.

Les mesures expérimentales du système contrôlé sont tracées dans la Figure 5.9 page 140. Dans ces conditions, la création de l'abstraction symbolique et la synthèse des contrôleurs a nécessité plus de deux jours de calculs sur un processeur de 3 GHz. Une précision supérieure de l'abstraction symbolique n'est donc pas envisageable pour ce système ne contenant pourtant que 4 variables d'état et 4 entrées de contrôle. Pourtant, la période d'échantillonnage τ (dont le choix est fortement lié à la précision de la partition de l'espace d'état) est légèrement trop grande, ce qui permet à des phénomènes non modélisés de s'accumuler et rend parfois le contrôleur incapable de réaliser les spécifications.

Contrôle symbolique compositionnel L'approche compositionnelle permet de résoudre le problème de la méthode centralisée décrite dans le paragraphe précédent. On choisit une décomposition du modèle global en 4 sous-systèmes de dimension 1, c'est à dire que pour chaque pièce $i \in \{1, 2, 3, 4\}$, on ne modélise que la température de la pièce et le contrôle de la ventilation associée : $I_i = I_i^c = J_i = i$ et $I_i^o = \emptyset$. Cette décomposition réduit très fortement le temps de calcul nécessaire puisque la complexité de cette méthode est devenue linéaire en α_x, α_u, et la dimension $n = 4$ de l'espace d'état. On peut alors créer une abstraction symbolique plus précise à moindre coût : on choisit $\alpha_x = 20$ et $\alpha_u = 9$ et l'intégralité des calculs sont réalisés en 1.1 s. L'augmentation de α_x nous permet aussi de réduire la période d'échantillonnage à $\tau = 10$ s pour diminuer l'influence des phénomènes thermiques non modélisés. Les mesures expérimentales du système contrôlé sont tracées dans la Figure 5.10 page 143, où l'on peut voir que les températures sont contrôlées dans leurs intervalles avec plus de facilité.

Il est à noter qu'aux vues du très faible temps de calcul nécessaire pour cette méthode, elle pourrait facilement être utilisée avec des précisions plus élevées ou appliquée à des systèmes de plus grande dimension. Bien qu'elle ne soit pas détaillée dans cette thèse, une décomposition alternative en sous-systèmes plus précis est possible avec des temps de calculs restant très réduits. Cette approche consiste à considérer un sous-système par pièce pour laquelle on contrôle la température et la ventilation ($I_i^c = J_i = i$) tout en modélisant aussi les variations de température des deux pièces voisines ($I_i^o = \mathcal{N}_i$) sans chercher à les contrôler. Les sous-systèmes obtenus ont donc 3 variables d'états mais ne gardent qu'une seule variable de contrôle, ce qui permet d'obtenir des complexités fortement réduites : dans les mêmes conditions que le cas centralisé ($\alpha_x = 10$ et $\alpha_u = 4$), le contrôleur est synthétisé en seulement 6 s au lieu de deux jours.

Conclusions

Le résumé de ce travail de thèse peut être conclu par quelques remarques et perspectives. Tout d'abord, comme cela a été mis en avant dans le chapitre applicatif sur le bâtiment intelligent, les résultats et stratégies de contrôle des autres chapitres peuvent se compléter pour obtenir un contrôle global. Ainsi, on peut commencer par utiliser la notion d'invariance contrôlée robuste pour choisir un intervalle dans lequel on souhaite maintenir l'état du système. La méthode de stabilisation robuste peut alors être appliquée à cet intervalle pour amener l'état à l'intérieur si le système a été initialisé à l'extérieur. Enfin, lorsque l'état est dans l'intervalle, on peut utiliser un contrôleur obtenu grâce aux méthodes symboliques pour garder l'état dans l'intervalle tout en optimisant un critère de performance. L'approche compositionnelle des méthodes symboliques nous permet de considérer ce contrôleur global pour des systèmes de grande dimension.

De nombreuses directions peuvent être données aux futurs développements des travaux présentés dans cette thèse. Pour les méthodes symboliques, on pourrait s'intéresser à d'autres spécifications de contrôle que la sûreté. Il serait également utile d'automatiser le choix de la période d'échantillonnage en fonction de la précision de la partition choisie. Enfin, dans l'approche compositionnelle, il serait intéressant de pouvoir modéliser deux types d'entrées de contrôle comme cela est déjà fait pour les états : des variables utilisées pour le contrôle et d'autres seulement observées pour augmenter la précision du modèle.

Le problème de passage à l'échelle de la méthode symbolique centralisée étant résolu grâce à l'approche compositionnelle, il est maintenant possible d'utiliser ces méthodes de synthèse de contrôleurs dans une structure similaire à la commande prédictive, où un contrôleur est synthétisé pour des valeurs de perturbations proches de la mesure actuelle et ce contrôleur est appliqué jusqu'à la prochaine mesure des perturbations. Cette structure de contrôle permettrait alors d'utiliser des abstractions plus précises dans une approche globale plus robuste. Il serait aussi possible de s'affranchir de la nécessité d'avoir un système coopératif en ne considérant que des comportements coopératifs locaux du système.

En ce qui concerne les applications, une première perspective est d'améliorer le modèle du bâtiment expérimental équipé de l'UFAD. Le contrôleur de ce système pourrait aussi être combiné à un estimateur des perturbations discrètes (portes et lampes) qui ont une forte influence sur les dynamiques. Cela permettrait alors de travailler sur un modèle plus précis et donc l'optimisation donnerait de meilleurs performances. Enfin, il serait intéressant d'appliquer ces méthodes de contrôles à d'autres types de système et notamment à des systèmes multi-agents pour lesquels l'approche compositionnelle est particulièrement adaptée.

Introduction

Green buildings Due to the rapidly growing population and the increasing demand in comfort, energy efficiency in buildings has become a major concern since buildings represent up to 40% of the total energy consumption in developed countries [PLOP08]. It has been shown that coordinating all control decisions affecting the indoor climate regulation (e.g. temperature, ventilation, light, humidity) could significantly reduce the global energy consumption of the building. Such approach requires the introduction of sensing, coordination and actuation capabilities in the building to measure the current situation, compute the most efficient control strategy at the building level and apply it locally in each room. The use of such technological and decision-making elements in a building constitutes the basic description of energetically efficient buildings, also known as intelligent or green buildings.

In this work, we focus on the temperature regulation in green buildings. This is a difficult problem due to the heterogeneous nature of the elements influencing the global behavior. Indeed, such dynamics combine both continuous behaviors (temperature variations following the first law of thermodynamics) and discrete transitions (e.g. a user entering a room or opening a window), which can best be described by the theory of hybrid systems. Due to these interactions, we cannot apply the classical methods from either continuous or discrete control theory and we need to use specific techniques adapted to the hybrid nature of the system.

Symbolic control The solution explored in this thesis to address this control problem is based on symbolic methods. The principle of such methods is to create a purely discrete abstraction of the original system, represented as a finite nondeterministic transition system and for which a controller can be more easily synthesized using methods in the field of discrete control. If we can prove some behavioral relationship (e.g. simulation, bisimulation, or their alternating and approximate versions [Tab09]) between the abstraction and the original model, then it means that all behaviors of the original system can be replicated on the abstraction. The alternating simulation relation also implies that a discrete controller synthesized on the abstraction can be refined into a controller of the original model that satisfies the same specifications. We thus talk about *hybrid control* since a discrete controller is applied to a continuous or hybrid system. Note that this name does not mean that this approach only applies to hybrid systems: it can be useful for any system whose dynamics are too complicated to be controlled with classical methods.

The name of symbolic method comes from the fact that the first step of creating the discrete abstraction is to partition the state space: each element of this partition

can be seen as a *symbol* representing all the continuous states it contains. Then, the transitions of the symbolic abstractions are obtained from a reachability analysis where we approximate the set of continuous states that can be reached (using a sampled version of the original system) from those in a given symbol. This approximation can be computed in several ways depending on the properties of the system, but the simplest one is obtained when the system satisfies a monotonicity property, described in the next paragraph.

Monotonicity Systems satisfying the monotonicity property can be found in a large variety of fields such as molecular biology, biochemical networks, population evolutions or thermal dynamics in buildings. A monotone system is defined as a system whose trajectories preserve some partial orderings on its state [AS03]. This means that if we consider an initial state x_0 *greater* than another one x'_0 and an input function $\mathbf{u}$ at all time *greater* than another one $\mathbf{u}'$, then the trajectory of the system starting on x_0 with the input function $\mathbf{u}$ always stays *above* the trajectory of the same system starting on x'_0 with the input function $\mathbf{u}'$. The subclass of cooperative systems corresponds to the case where we use the classical componentwise inequalities as our partial orderings:

$$x_0 \geq x'_0,\ \forall t \in \mathbb{R}_0^+,\ \mathbf{u}(t) \geq \mathbf{u}'(t) \Rightarrow \forall t \in \mathbb{R}_0^+,\ \mathbf{x}(t; x_0, \mathbf{u}) \geq \mathbf{x}(t; x'_0, \mathbf{u}').$$

This property is particularly useful to bound any trajectory of the system by two particular trajectories that involve the extremal values (with respect to the chosen partial orderings) of the state and input variables. Thus, to create the symbolic abstraction, if the symbols are taken as multi-dimensional intervals of the state space, a tight over-approximation of the reachable set can be obtained simply by computing two successors of the sampled system: one for the lower bound of the symbol considered and one for its upper bound.

Controlled invariance In the control problem of regulating the temperature in a building, each user specifies a temperature setpoint corresponding to the comfort he demands for his room. Since we consider a system subject to unknown but bounded disturbances, classical stability may not be achieved and we need to relax these comfort specifications into intervals of temperatures around these setpoints. Therefore, realizing the global comfort specifications corresponds to finding a control strategy that maintains the state (vector of all temperatures in the building) in a multi-dimensional interval despite the adversarial behavior of the environment. In this work, this notion is referred to as a safety game for discrete-time systems (such as the symbolic abstraction) and robust controlled invariance for continuous-time systems.

Since the invariance objective is in a vector interval which naturally has lower and upper bounds, the monotonicity property can also be useful to characterize the notion of robust controlled invariance. Although this characterization describes the ability to control the system in a set rather than defining an actual control strategy, it provides interesting elements for comparison with the symbolic methods and facilitates the choice of the interval considered in the abstraction task.

Main contributions

Apart from Chapter 1 where we describe the class of systems considered and particularly the notions of monotone and cooperative systems, the contributions of this thesis are organized in four chapters. Chapter 2 introduces the notion of robust controlled invariance as a preliminary result on robust control in a set for comparison with the control strategies in the next chapters. A controller synthesis based on a symbolic approach is given in Chapter 3 with the dual objective of keeping the state in some safety bounds and optimizing a performance criterion. A compositional approach to this problem is presented in Chapter 4 to solve its scalability issue by synthesizing partial controllers on partial descriptions of the system. Finally, a validation of the control strategies obtained in the previous three chapters is given in Chapter 5 on the temperature regulation for an experimental small-scale building. The main results of these chapters are summarized below.

Chapter 2: Robust controlled invariance

In this chapter, we consider a control system subject to disturbances

$$\dot{x} = f(x, u, w), \tag{1.1}$$

where x, u and w are the state, control input and disturbance input, respectively. This system is assumed to be cooperative as in Definition 1.5, with bounded inputs $u \in [\underline{u}, \overline{u}]$ and $w \in [\underline{w}, \overline{w}]$.

We first describe the notion of *robust invariance* (robustness with respect to both control and disturbance inputs) and characterize in Theorem 2.3 the minimal *robust invariant* interval: the state always remains above the equilibrium obtained with the constant inputs $\mathbf{u} = \underline{u}$ and $\mathbf{w} = \underline{w}$ and below the equilibrium defined by $\mathbf{u} = \overline{u}$ and $\mathbf{w} = \overline{w}$, if these equilibria exist.

The *robust controlled invariance* defines a set such that there exists a feedback controller maintaining the state in this set for any value of the disturbance input w. With the additional local control property (Definition 1.12), stating that each component of the control input u only directly influence a single state variable in the vector field (1.1), a robust controlled invariant interval $[\underline{x}, \overline{x}]$ is characterized by the sign of the vector field (1.1) and using only the extremal values of its variables:

$$\begin{cases} f(\overline{x}, \underline{u}, \overline{w}) \leq 0, \\ f(\underline{x}, \overline{u}, \underline{w}) \geq 0. \end{cases} \tag{Theorem 2.5}$$

The first equation means that on the upper bound $\overline{x}$ of the interval and with the maximal disturbance $\overline{w}$, the minimal value of the control input $\underline{u}$ can force a decrease on all state variables ($f \leq 0$, with a componentwise inequality). Similarly for the second equation, the maximal control can force an increase of the state when it is on the lower bound of the interval with the minimal disturbance. If both conditions are satisfied, we know that the control input u can maintain the state x in the interval $[\underline{x}, \overline{x}]$ for any condition of the disturbance w.

Next, Theorem 2.8 studies the case when the previous theorem is satisfied with strict inequalities and on an interval reduced to a single point $\underline{x} = \overline{x} = x^*$. These

conditions mean that there exists a feedback controller that can maintain the state of the system in any small robust controlled invariant interval around x^* and equivalently, in any small neighborhood of x^*. The state x^* is thus said to be *robustly locally stabilizable*.

While the robust controlled invariance describes the ability to keep the state in a set, the *robust set stabilization* introduced at the end of the chapter corresponds to the ability to bring the state of the system in a set and in finite time when it is initialized outside. Theorem 2.11 proves that such stabilizing controller can be obtained if the target set is a robust controlled invariant interval. The method used for this robust set stabilization consists in considering a family of robust controlled invariant intervals which is decreasing with respect to the set inclusion: in each of these intervals we can force the state toward its interior until it reaches the final interval. In Section 2.5.2, we also provide three possible definitions of such decreasing families of intervals.

Chapter 3: Symbolic control

In this chapter and the next one, we start from a sampled version of the continuous dynamics (1.1), denoted as S and described as a transition system. The sampled system S is abstracted into a finite transition system S_a whose set of inputs is a discretization of the control input interval $[\underline{u}, \overline{u}]$ and the set of states is a partition of the state space into identical intervals called symbols. Let τ denote the sampling period of S and $\Phi(\tau, x_0, \mathbf{u}, \mathbf{w})$ the state reached at time τ from the state x_0 and with input functions $\mathbf{u}$ and $\mathbf{w}$. The monotonicity property provides a tight over-approximation of the reachable set of S from any continuous state in a symbol $s = [\underline{s}, \overline{s}]$ and with a constant input u:

$$\forall x \in s,\ \mathbf{w} : [0, \tau] \to [\underline{w}, \overline{w}],\ \Phi(\tau, x, u, \mathbf{w}) \in [\Phi(\tau, \underline{s}, u, \underline{w}), \Phi(\tau, \overline{s}, u, \overline{w})]. \tag{3.5}$$

The transitions of S_a can thus be defined using (3.5): the successors of a symbol s with input u are all the symbols intersecting the over-approximation in (3.5). Note that this method is not new and can be seen, e.g. in [MR02].

From the use of the over-approximation (3.5) to define S_a, it is shown in Proposition 3.6 that an alternating simulation relation exists between S_a and S. With our particular construction of S_a, this relationship states that for any symbol s and continuous state $x \in s$, an input u chosen in S_a implies that any successor of x with input u in S belongs to a symbol successor of s with input u in S_a. As a consequence, if a controller of S_a is synthesized to realize some specification on the symbols, it can be refined into a controller of S realizing the same specification.

The safety specification for S is to stay in a safe interval $[\underline{x}, \overline{x}]$ and the corresponding safety for S_a is to stay in the symbols that are contained in this interval. Using a classical fixed-point algorithm, we synthesize a safety controller for S_a by forbidding the inputs possibly leading outside of the interval. This controller is refined into a controller for S, which is then proven to also realize the safety specification on S (Theorem 3.7). In Example 3.2 and Figure 3.4, we show that for an interval which is not robust controlled invariant, the safe subset obtained from the symbolic method is larger than the maximal robust controlled invariant sub-interval that could be obtained from the results of Chapter 2.

Since several safe control strategies may be allowed after the previous step, we choose one optimizing the following performance criterion:

$$\sum_{k=0}^{+\infty} \lambda^k g(x^k, u^k), \tag{3.1}$$

where $(x^0, u^0, x^1, u^1, \dots)$ is an infinite trajectory of the controlled system S, $g(x, u)$ is the cost of using input u from the state x and $\lambda \in (0, 1)$ is a discount factor that reduces the influence of the future steps. In S_a, the exact state x is unknown and we thus introduce the cost function $g_a(s, u) = \max_{x \in s} g(x, u)$. If we choose $N \in \mathbb{N}$ sufficiently large, we can approximate (3.1) by

$$\sum_{k=0}^{N} \lambda^k g_a(s^k, u^k), \tag{3.11}$$

where $x^k \in s^k$ for all k. We thus minimize the accumulated cost (3.11) using a dynamic programming algorithm over a finite horizon of N sampling periods. Note that due to the non-determinism induced by the abstraction and the unknown disturbances, this is a worst-case optimization. Then, a receding horizon control scheme is applied to the result of this optimization to obtain a deterministic controller. In Theorem 3.10, we provide a performance guarantee as an upper bound on the performance criterion (3.1) for any infinite trajectory of S controlled with the receding horizon controller.

Chapter 4: Compositional symbolic control

To address the scalability issue of the centralized symbolic method from Chapter 3, we introduce a compositional approach where the previously described symbolic method is applied to several subsystems that partially describe the global dynamics. To define the subsystem of index i, we introduce six index sets: I_i, I_i^c, I_i^o and K_i for state dimensions and J_i and L_i for control input dimensions. Their roles can be summarized as follows: in the i^{th} subsystem, we model the states $x_{I_i} = (x_{I_i^c}, x_{I_i^o})$ where $x_{I_i^c}$ are to be controlled using the inputs u_{J_i} and $x_{I_i^o}$ are simply observed to increase the precision of the subsystem while x_{K_i} and u_{L_i} are unobserved and considered as external disturbances. We assume that over all subsystems, each state component appears exactly once as a controlled state (in $x_{I_i^c}$, for some i). Similarly, each input component appears exactly once as a controlled input (in some u_{J_i}).

The symbolic abstraction S_i of subsystem i is obtained under two assume-guarantee obligations, where we consider that, when possible, other subsystems realize the safety specifications on the other state components:

$$\text{Unobserved states: } x_{K_i} \in [\underline{x}_{K_i}, \overline{x}_{K_i}], \qquad \text{(A/G Obligation 1)}$$

$$\text{Observed but uncontrolled states: } x_{I_i^o} \in [\underline{x}_{I_i^o}, \overline{x}_{I_i^o}]. \qquad \text{(A/G Obligation 2)}$$

Due to the loss of information (states x_{K_i} and inputs u_{L_i}) and the use of A/G Obligation 1, the over-approximation of the reachable set of subsystem i from a symbol s_i and with an input u_i is more conservative than (3.5):

$$[\Phi(\tau, (\underline{s_i}, \underline{x}_{K_i}), (u_i, \underline{u}_{L_i}), \underline{w}), \Phi(\tau, (\overline{s_i}, \overline{x}_{K_i}), (u_i, \overline{u}_{L_i}), \overline{w})]. \tag{4.2}$$

The symbolic abstraction S_i is then obtained using the method of the previous chapter combined with the over-approximation (4.2) and A/G Obligation 2 that removes some unsafe transitions in S_i when they are only due to the uncontrolled state components $x_{I_i^o}$. The controller syntheses on S_i for safety and performance optimization are done as in Chapter 3.

We first prove in Proposition 4.1 that the composition of all subsystems is alternatingly simulated by the symbolic abstraction S_a from Chapter 3. Then, with the transitivity of the alternating simulation relation, the composition of the subsystems is also alternatingly simulated by the original system S.

With this alternating simulation, we show in Theorem 4.3 that the controller obtained from composing the safety controllers of all subsystems realizes the safety specification of S. As expected, the safe set obtained from the less accurate compositional method is included in the one obtained in Chapter 3 (Corollary 4.4).

Theorem 4.5 then provides performance guarantees on the original system S controlled with the composition of the receding horizon controllers obtained on the subsystems. We compare the performance guarantees in Corollary 4.9 and show that the total cost (3.1) of a trajectory of S has a tighter upper bound when controlled with the controller from the centralized method (Theorem 3.10) than with the controller from the compositional approach (Theorem 4.5).

Hence, for both safety and performance, we thus obtain similar results than with the centralized method in Chapter 3, though naturally weaker due to the less accurate models involved. On the other hand, these results come with a significant reduction of the computational complexity, discussed in Section 4.6 and Table 4.3.

Chapter 5: Control in intelligent buildings

The last chapter aims at providing an experimental evaluation of the results from Chapters 2 to 4 on the temperature regulation in a 4-room small-scale building. The temperature control in each room is done with a cold air inflow forced by a controlled fan at the floor level. Assuming a uniform temperature in each room, the temperature variations are derived from the energy conservation equation (first law of thermodynamics) and the mass conservation equation in this room. The obtained physical model in Section 5.2.2 contains four types of heat transfers: thermal conduction through the walls, radiation from heat sources, cold air inflow from the ventilation and mass flow rate through open doors. This model is evaluated in Section 5.2.3 using a gray-box identification procedure to keep the general form of the dynamics while abstracting their physical meaning. The model properties required to apply the control methods from the previous chapters, particularly the monotonicity, are proven in Section 5.3.

The robust controlled invariance and robust set stabilization are combined and validated in Section 5.4. The robust controlled invariance is realized with a simple decentralized affine controller:

$$\mathsf{u}_i(x) = \underline{u_i} + (\overline{u_i} - \underline{u_i})\frac{\overline{x_i} - x_i}{\overline{x_i} - \underline{x_i}}. \tag{2.4}$$

In Figure 5.8, we provide another experiment for which we give a detailed description on how the robust set stabilization is realized.

Using Theorem 2.5 to choose a robust controlled invariant interval, the centralized symbolic method from Chapter 3 is applied to the system in Section 5.5.1. The final controller is synthesized from the optimization involving the following cost function:

$$g_a(s^k, u^k, u^{k-1}) = \frac{\|u^k\|}{\|\overline{u} - \underline{u}\|} + \frac{\|u^k - u^{k-1}\|}{\|\overline{u} - \underline{u}\|} + \frac{\|s_*^k - T_*\|}{\|(\overline{T} + \underline{T})/2\|}. \tag{5.14}$$

This function makes a tradeoff between the minimization of three criteria: the control, the variations of the control and the distance between the center s_*^k of the current symbol s^k and the center T_* of the interval. All these costs are normalized to give them equal weights. The main drawback of this method clearly appears on this application: for this system with only 4 states and 4 control inputs, the symbolic abstraction and controller synthesis with the low precision of 10 symbols and 4 inputs per dimension take more than two days of computation.

Then, the compositional approach from Chapter 4 is applied in Section 5.5.2. We consider a decomposition of the dynamics into 4 subsystems, each modeling and controlling a single temperature with the fan control of the same room. The four abstractions and controllers are computed in a few seconds even for very high precision (e.g. 0.02 °C when we take 200 symbols per dimension).

Finally, in Section 5.6, we briefly discuss the possibilities to combine the main control strategies presented in this work. We thus can use the robust controlled invariance to choose an interval, the robust set stabilization to bring the state into this interval and the symbolic control for a more efficient strategy when the state is in the interval.

Publications

The work presented in this thesis led to several papers either published or submitted.

Journal paper:

- P.-J. Meyer, A. Girard and E. Witrant, Robust controlled invariance for monotone systems: application to ventilation regulation in buildings. Provisionally accepted in *Automatica*.

International conference:

- P.-J. Meyer, A. Girard and E. Witrant, Safety control with performance guarantees of cooperative systems using compositional abstractions. Submitted at the 5^{th} *IFAC Conference on Analysis and Design of Hybrid Systems*, 2015.
- P.-J. Meyer, H. Nazarpour, A. Girard and E. Witrant, Experimental Implementation of UFAD Regulation based on Robust Controlled Invariance. 13^{th} *European Control Conference*, Strasbourg, France, pp. 1468-1473, 2014.
- P.-J. Meyer, A. Girard and E. Witrant, Controllability and invariance of monotone systems for robust ventilation automation in buildings. 52^{nd} *IEEE Conference on Decision and Control*, Florence, Italy, pp. 1289-1294, 2013.

Conference "poster abstract":

- P.-J. Meyer, A. Girard and E. Witrant, Poster: Symbolic Control of Monotone Systems, Application to Ventilation Regulation in Buildings. 18^{th} *ACM International Conference on Hybrid Systems: Computation and Control*, Seattle, USA, pp. 281-282, 2015.

- P.-J. Meyer, H. Nazarpour, A. Girard and E. Witrant, Poster abstract: Robust Controlled Invariance for UFAD Regulation. 5^{th} *ACM Workshop on Embedded Systems For Energy-Efficient Buildings (BuildSys)*, Rome, Italy, pp. 1-2, 2013.

Chapter 1

Monotone control system

In this chapter, we present the class of systems that is considered throughout this thesis and list the assumptions required by some of the results in the next chapters. At the end of the chapter, we introduce some simple systems satisfying all these assumptions that will be used to illustrate the main results in Chapters 2 to 4. Let us first introduce some notations. We are interested in continuous-time control systems subject to disturbances and described by the non-linear ordinary differential equation:

$$\dot{x} = f(x, u, w), \tag{1.1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$ and $w \in \mathbb{R}^q$ denote the state, the control input and the disturbance input, respectively. The function $f : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^n$ is the vector field describing the dynamics of the system. The trajectories of (1.1) are denoted $\Phi(\cdot, x_0, \mathbf{u}, \mathbf{w})$ where $\Phi(t, x_0, \mathbf{u}, \mathbf{w})$ is the state reached at time $t \in \mathbb{R}_0^+$ from the initial state $x_0 \in \mathbb{R}^n$, under control and disturbance inputs $\mathbf{u} : \mathbb{R}_0^+ \to \mathbb{R}^p$ and $\mathbf{w} : \mathbb{R}_0^+ \to \mathbb{R}^q$. When the control inputs of system (1.1) are generated by a state-feedback controller $\mathsf{u} : \mathbb{R}^n \to \mathbb{R}^p$, the dynamics of the closed-loop system is given by:

$$\dot{x} = f_{\mathsf{u}}(x, w) = f(x, \mathsf{u}(x), w), \tag{1.2}$$

and its trajectories are denoted as $\Phi_{\mathsf{u}}(\cdot, x_0, \mathbf{w})$.

1.1 Monotonicity

While the initial work by Müller [Mül27], Kamke [Kam32] and Krasnoselskii [Kra68] on comparison arguments in differential equations placed the first stones of what would then become the theory of monotone systems, the development of this topic for continuous-time systems was mainly contributed by Hirsch and Smith. Among their most notable publications, we could cite the book [Smi95], the more recent book chapter [HS05] or the survey [Smi88]. Their main results on monotone autonomous systems are recalled in Section 1.1.1. These results were extended to systems with inputs by Angeli and Sontag [AS03] and are presented in Section 1.1.2 in a formulation matching the definition of our system (1.1).

Systems satisfying the monotonicity property and in particular the subclass of cooperative systems have been used in a large variety of fields such as molecular

biology and biochemical networks [Son07, BG13], population evolutions [DLAS05], or thermal dynamics in buildings, which is the application considered in Chapter 5. The potential scope of application of the results on monotone systems has also been widened by the notion of mixed-monotonicity [GH94] that can be applied to both continuous-time [ESS06] and discrete-time systems [Smi06]. Indeed, it is shown in these works that some non-monotone system can be decomposed into its increasing and decreasing parts. As a result, if we create a new system from the duplication of the dynamics of a mixed-monotone system, this new system is monotone.

1.1.1 Autonomous systems

An autonomous dynamical system $\dot{x} = f(x)$ is said to be monotone when its trajectories Φ preserve some suitable partial ordering on the state. Simply put, if an initial state x_0 is "greater" than another x_0', then the trajectory of this monotone system starting from x_0 always stays "above" the trajectory starting from x_0'. These quoted terms of comparison on the states are linked to the notion of partial ordering which is defined below. In the case of a system with a single state variable using the classical comparison $\geq$, the above sentence describing monotonicity can be illustrated as in Figure 1.1.

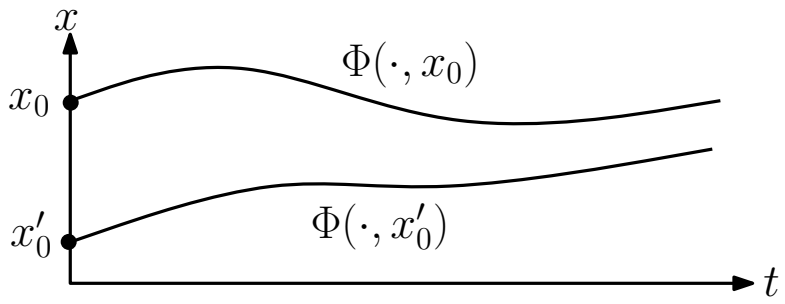


Figure 1.1 – Monotonicity illustrated on a scalar autonomous system.

For systems with more than one state or more complicated comparison relations, we need to introduce a general definition of the partial orderings. A partial ordering on a set X is a relation $\succeq$ which is:

- reflexive: $x \succeq x$ for all x in X;
- transitive: $x \succeq y$ and $y \succeq z$ implies $x \succeq z$;
- antisymmetric: $x \succeq y$ and $y \succeq x$ implies $x = y$.

In [Smi95], the partial orderings are defined on a Banach space X associated with a positive cone K with the following properties:

- cone: $\alpha K \subseteq K$ for all positive α;
- convex: $K + K \subseteq K$;
- pointed: $K \cap (-K) = \{0\}$.

This cone thus allows the following definition for the partial ordering on X:

$$x \succeq x' \Leftrightarrow x - x' \in K$$

and a stricter formulation

$$x \gg x' \Leftrightarrow x - x' \in int(K),$$

where $int(K)$ is the interior of the cone. We denote indifferently $x \succeq x'$ and $x' \preceq x$, $x \gg x'$ and $x' \ll x$. Given $\underline{x}$ and $\overline{x}$ in X with $\overline{x} \succeq \underline{x}$, $[\underline{x}, \overline{x}]$ denotes the interval such that:

$$x \in [\underline{x}, \overline{x}] \Leftrightarrow \overline{x} \succeq x \succeq \underline{x}. \tag{1.3}$$

Similarly for an open interval, $x \in (\underline{x}, \overline{x})$ if and only if $\overline{x} \gg x \gg \underline{x}$.

Consider an autonomous system defined similarly to (1.1) but without inputs. The system is said to be monotone if its trajectories preserve a partial ordering on the states.

Definition 1.1 (Monotonicity)**.** *The system $\dot{x} = f(x)$ with trajectories Φ is monotone with respect to the partial ordering $\succeq$ if the following implication holds:*

$$x \succeq x' \Rightarrow \forall t \geq 0,\ \Phi(t, x) \succeq \Phi(t, x').$$

In most applications, the considered state space X is the Euclidean space $\mathbb{R}^n$ and the cone K inducing the partial ordering is an orthant of $\mathbb{R}^n$. A particular case has a special name: cooperative system.

Definition 1.2 (Cooperative system)**.** *A system is cooperative when it is monotone with respect to the partial ordering induced by the positive orthant.*

With a simple change of variables, a system which is monotone with respect to any orthant of $\mathbb{R}^n$ can easily be replaced by one using the positive orthant $\mathbb{R}^n_+$. Thus, we focus our considerations on cooperative systems only. This implies that the resulting partial ordering $\succeq$ corresponds to the classical componentwise comparison:

$$x \succeq x' \Leftrightarrow \forall i \in \{1, \ldots, n\},\ x_i \geq x'_i.$$

Similarly, the stricter formulation $x \gg x'$ is equivalent to checking $x_i > x'_i$ for all indices i.

Since the actual trajectory Φ of a system is rarely known, proving its monotonicity from Definition 1.1 is not possible. Instead, we look for new characterization of monotone systems in terms of their vector fields. Firstly, cooperative systems have been characterized by the Kamke-Müller condition [Kam32, Mül27] as follows.

Proposition 1.3. *The system $\dot{x} = f(x)$ with locally Lipschitz vector field f is cooperative if and only if the following implication holds for all $i \in \{1, \ldots, n\}$:*

$$x \succeq x',\ x_i = x'_i \Rightarrow f_i(x) \geq f_i(x').$$

In the particular case of a system with a continuously differentiable vector field, the condition in Proposition 1.3 can be replaced by one in terms of the partial derivatives of the vector field [Smi95].

Proposition 1.4. *The system $\dot{x} = f(x)$ with continuously differentiable vector field f is cooperative if and only if for all $i \in \{1, \dots, n\}$ we have:*

$$\forall j \neq i,\ \forall x \in \mathbb{R}^n,\ \frac{\partial f_i}{\partial x_j}(x) \geq 0.$$

In the case where we would want to prove the monotonicity of a system with respect to the partial ordering induced by another orthant K, the conditions to be checked on the partial derivatives of the vector field would be of the form

$$(-1)^{\varepsilon_i + \varepsilon_j} \frac{\partial f_i}{\partial x_j}(x) \geq 0,$$

where $\varepsilon_k = 0$ if the projection of the cone K on the k^{th} dimension is the positive half axis and $\varepsilon_k = 1$ if it is the negative half axis.

1.1.2 Systems with inputs

The classical results presented in Section 1.1.1 were extended by Angeli and Sontag to systems with inputs [AS03]. Thus, we present how the definitions and propositions of the previous section have to be modified to characterize the monotonicity of the system (1.1) with both control and disturbance inputs.

Let us first complete the illustration example from Section 1.1.1 to get an intuitive idea of the notion of monotonicity with respect to inputs. If an input function $\mathbf{u}$ is at all time "greater" than another $\mathbf{u}'$, then the trajectory with $\mathbf{u}$ is always "above" the one with $\mathbf{u}'$, assuming both trajectories start from the same initial state. This notion is illustrated in Figure 1.2 for a system with a single state and a single input. For clarity of this illustration, we focus on the preservation of the input partial ordering only (by taking the same initial state), but it can of course be combined with the preservation of the state partial ordering from Figure 1.1.

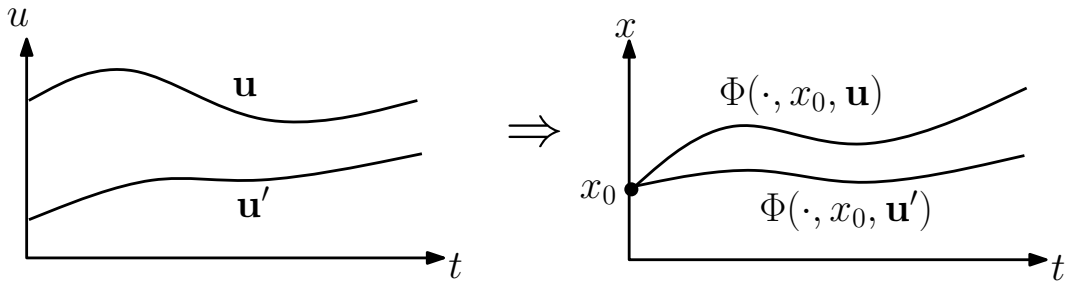


Figure 1.2 – Monotonicity illustration on a scalar system with an input.

Since the trajectory $\Phi(\cdot, x_0, \mathbf{u}, \mathbf{w})$ of (1.1) uses the functions $\mathbf{u} : \mathbb{R}_0^+ \to \mathbb{R}^p$ and $\mathbf{w} : \mathbb{R}_0^+ \to \mathbb{R}^q$, we need to extend the definition of the partial ordering to functions of time:

$$\mathbf{u} \succeq_u \mathbf{u}' \Leftrightarrow \forall t \geq 0,\ \mathbf{u}(t) \succeq_u \mathbf{u}'(t).$$

We can give a similar definition for $\mathbf{w} \succeq_w \mathbf{w}'$ with $\mathbf{w}, \mathbf{w}' : \mathbb{R}_0^+ \to \mathbb{R}^q$. Note that to avoid confusion, we now differentiate the partial orderings $\succeq_x$ on the state space

$\mathbb{R}^n$, $\succeq_u$ on the control input space $\mathbb{R}^p$ or the set of control input functions $\mathbb{R}_0^+ \to \mathbb{R}^p$ and $\succeq_w$ on the disturbance input space $\mathbb{R}^q$ or the set of disturbance input functions $\mathbb{R}_0^+ \to \mathbb{R}^q$. We similarly extend the interval notation to functions: if $\overline{u} \succeq_u \underline{u}$, we note $\mathbf{u} \in [\underline{u}, \overline{u}]$ if and only if $\mathbf{u}(t) \in [\underline{u}, \overline{u}]$ for all $t \geq 0$.

We can now present the extension of the previous definitions and propositions to systems with inputs [AS03].

Definition 1.5 (Monotonicity)**.** *The system (1.1) is monotone with respect to the partial orderings $\succeq_x$, $\succeq_u$ and $\succeq_w$ if the following implication holds:*

$$x \succeq_x x',\ \mathbf{u} \succeq_u \mathbf{u}',\ \mathbf{w} \succeq_w \mathbf{w}' \Rightarrow \forall t \geq 0,\ \Phi(t, x, \mathbf{u}, \mathbf{w}) \succeq_x \Phi(t, x', \mathbf{u}', \mathbf{w}').$$

If in addition the partial orderings $\succeq_x$, $\succeq_u$ and $\succeq_w$ are induced by the positive orthants $\mathbb{R}^n_+$, $\mathbb{R}^p_+$ and $\mathbb{R}^q_+$ respectively, (1.1) is cooperative.

Next is the extension of the Kamke-Müller condition.

Proposition 1.6. *The system (1.1) with locally Lipschitz vector field f is cooperative if and only if the following implication holds for all $i \in \{1, \ldots, n\}$:*

$$x \succeq_x x',\ x_i = x'_i,\ u \succeq_u u',\ w \succeq_w w' \Rightarrow f_i(x, u, w) \geq f_i(x', u', w').$$

Then, the characterization of cooperative systems in terms of partial derivatives of the vector field.

Proposition 1.7. *The system (1.1) with continuously differentiable vector field f is cooperative if and only if for all $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$, $w \in \mathbb{R}^q$, $i, j \in \{1, \ldots, n\}$, $j \neq i$, $k \in \{1, \ldots, p\}$, $l \in \{1, \ldots, q\}$:*

$$\frac{\partial f_i}{\partial x_j}(x, u, w) \geq 0,\ \frac{\partial f_i}{\partial u_k}(x, u, w) \geq 0,\ \frac{\partial f_i}{\partial w_l}(x, u, w) \geq 0.$$

In the original versions of Propositions 1.3 and 1.4 from [Smi95] or Propositions 1.6 and 1.7 from [AS03], the vector field f is initially defined on an open subset of $\mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q$. For the propositions to hold when the variables x, u and w are defined on closed sets, these sets are required to satisfy the approximability property defined below.

Definition 1.8 (Approximability)**.** *Let $z \in \{x, u, w\}$ be defined on a closed set Z. Let $int(Z)$ be the interior of Z. If for all $z_a \succeq_z z_b$ there exists sequences $\{z_a^i\}, \{z_b^i\} \in int(Z)$ such that $z_a^i \underset{i\to\infty}{\longrightarrow} z_a$, $z_b^i \underset{i\to\infty}{\longrightarrow} z_b$ and $z_a^i \succeq_z z_b^i$ for all i, then Z satisfies the approximability property.*

In this thesis, we consider that the state x can always be defined in a large enough open set in $\mathbb{R}^n$ for which Definition 1.8 is not required. In addition, to exploit the advantages of the monotonicity detailed later in this chapter, all continuous inputs are considered to vary in multi-dimensional intervals as defined in (1.3). These intervals are created either to constrain control inputs or to give the forecasted range of disturbances. The approximability is thus satisfied for these intervals since convexity of the set is a sufficient condition for this property [AS03].

1.1.3 Systems with discrete inputs

Here, we consider a system of the form $\dot{x} = f(x, \delta)$ with a discrete input $\delta \in \Delta$, where Δ is a discrete subset of $\mathbb{R}^q$. Let $\succeq_\delta$ be the classical componentwise partial ordering on $\mathbb{R}^q$. In this system, the approximability property from Definition 1.8 cannot be satisfied for δ defined on a discrete set Δ whose interior is empty: $int(\Delta) = \emptyset$. We thus need to show that the monotonicity with respect to a discrete input can also be characterized by a condition similar to the one from Kamke and Müller for continuous inputs (Proposition 1.6).

Proposition 1.9. *The system $\dot{x} = f(x, \delta)$ with f locally Lipschitz in x is cooperative if and only if the following implication holds for all $i \in \{1, \dots, n\}$:*

$$x \succeq_x x',\ x_i = x'_i,\ \delta \succeq_\delta \delta' \Rightarrow f_i(x, \delta) \geq f_i(x', \delta').$$

Proof. We consider the case where there is a single discrete input δ ($\Delta \subseteq \mathbb{R}$). The general case can easily be extended by proving the monotonicity for each input separately. Since Proposition 1.3 already gives the conditions for the monotonicity with respect to the state x, here we only focus on the monotonicity with respect to δ. Let g be the function defined below and regarded as a continuous extension of f between two values $\delta > \delta' \in \Delta$ of its discrete input:

$$g(x, d, \delta, \delta') = \frac{d - \delta'}{\delta - \delta'} f(x, \delta) + \frac{d - \delta}{\delta' - \delta} f(x, \delta'), \quad d \in [\delta', \delta].$$

We can thus see that g mimics the dynamics of f with $g(x, \delta, \delta, \delta') = f(x, \delta)$ and $g(x, \delta', \delta, \delta') = f(x, \delta')$ while being differentiable in its continuous variable $d \in [\delta', \delta]$:

$$\frac{\partial g}{\partial d}(x, d, \delta, \delta') = \frac{f(x, \delta) - f(x, \delta')}{\delta - \delta'}.$$

Let Φ_f and Φ_g denote the trajectories of the dynamical systems $\dot{x} = f(x, \delta)$ and $\dot{x} = g(x, d, \delta, \delta')$ respectively.

For the sufficient condition, assume that

$$\delta \geq \delta' \in \Delta \Rightarrow f(x, \delta) \succeq_x f(x, \delta'). \tag{1.4}$$

Let $\boldsymbol{\delta} \geq \boldsymbol{\delta}'$ be two input functions in $\mathbb{R}_0^+ \to \Delta$ and partition the time domain $\mathbb{R}_0^+$ into intervals $I_i = [t_i, t_{i+1})$ where $\boldsymbol{\delta}$ and $\boldsymbol{\delta}'$ are constant. As illustrated in Figure 1.3 in an example where the functions $\boldsymbol{\delta} \geq \boldsymbol{\delta}'$ take their values in a binary domain $\Delta = \{0, 1\}$, this partition of $\mathbb{R}_0^+$ means that the switches of $\boldsymbol{\delta}$ and $\boldsymbol{\delta}'$ only happen on the instants t_i and we can introduce the constant values δ_i and δ'_i such that:

$$\forall t \in [t_i, t_{i+1}),\ \boldsymbol{\delta}(t) = \delta_i,\ \boldsymbol{\delta}'(t) = \delta'_i.$$

Since we already assume that the system is cooperative with respect to the state, if $\delta_i = \delta'_i$ on $I_i = [t_i, t_{i+1})$, then the following implication holds:

$$\Phi_f(t_i, x, \delta_i) \succeq_x \Phi_f(t_i, x, \delta'_i) \Rightarrow \forall t \in I_i,\ \Phi_f(t, x, \delta_i) \succeq_x \Phi_f(t, x, \delta'_i). \tag{1.5}$$

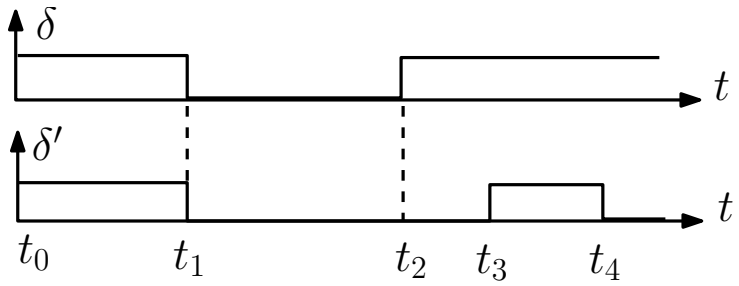


Figure 1.3 – $\mathbb{R}_0^+$ partition illustration for $\boldsymbol{\delta} \geq \boldsymbol{\delta}' : \mathbb{R}_0^+ \to \{0, 1\}$.

If $\delta_i > \delta_i'$ on I_i, then $g(x, d, \delta_i, \delta_i')$ is well-defined, (1.4) gives $\frac{\partial g}{\partial d} \succeq_x 0$ and g is cooperative with respect to its continuous input d. Applying Definition 1.5 for g with $\mathbf{d}(t) = \delta_i$ and $\mathbf{d}'(t) = \delta_i'$ for all $t \in I_i$, we have:

$$\Phi_f(t_i, x, \delta_i) \succeq_x \Phi_f(t_i, x, \delta_i') \Rightarrow \forall t \in I_i,\ \Phi_f(t, x, \delta_i) \succeq_x \Phi_f(t, x, \delta_i'). \tag{1.6}$$

Equations (1.5) and (1.6) cover all possible situations of $\boldsymbol{\delta} \geq \boldsymbol{\delta}'$. Combined with the fact that the initial state is independent of the inputs ($\Phi_f(t_0, x, \boldsymbol{\delta}) = \Phi_f(t_0, x, \boldsymbol{\delta}') = x$), we obtain the monotonicity with respect to the discrete input as in Definition 1.5.

We prove the necessary condition by contradiction. Assume that we have:

$$\boldsymbol{\delta} \geq \boldsymbol{\delta}' : \mathbb{R}_0^+ \to \Delta \Rightarrow \forall t \geq 0,\ \Phi_f(t, x, \boldsymbol{\delta}) \succeq_x \Phi_f(t, x, \boldsymbol{\delta}'), \tag{1.7}$$

and that there exists $\delta \geq \delta' \in \Delta$ and $x \in \mathbb{R}^n$ such that $f(x, \delta) \not\succeq_x f(x, \delta')$. Thus there exists $i \in \{1, \ldots, n\}$ such that $f_i(x, \delta) < f_i(x, \delta')$ and since $\Phi_f(0, x, \delta) = \Phi_f(0, x, \delta') = x$, there exists $\varepsilon > 0$ such that $\Phi_{f,i}(\varepsilon, x, \delta) < \Phi_{f,i}(\varepsilon, x, \delta')$, which contradicts the first assumption (1.7). Hence, (1.7) and $\delta \geq \delta' \in \Delta$ implies $f(x, \delta) \succeq_x f(x, \delta')$. □

This result will be useful in the application of Chapter 5 where the considered system is subject to both continuous and binary disturbances.

1.1.4 Time-dependent vector fields

In this thesis, we focus on systems of the form (1.1) where the vector field f is time-independent. The definitions and results in Sections 1.1.1 through 1.1.3 have been given for such systems. However, it is worth noting that all these results are still valid in their current form for time-dependent vector fields [HS05].

In particular, this alternative definition can be useful when the monotonicity or cooperativeness is not satisfied (or not required) with respect to one of the input variables. As the results of Chapter 3 do not require the system (1.1) to be monotone with respect to its control input u, we present the new definitions in this particular case. We consider that u is a function of time and define the alternative system with vector field $F : \mathbb{R}_0^+ \times \mathbb{R}^n \times \mathbb{R}^q \to \mathbb{R}^n$:

$$\dot{x} = F(t, x, w), \text{ with } F(t, x, w) = f(x, u(t), w). \tag{1.8}$$

Definition 1.5 applied to (1.8) thus can be written without a partial ordering on the control input.

Definition 1.10 (Monotonicity)**.** *The system (1.8) is monotone with respect to the partial orderings $\succeq_x$ and $\succeq_w$ if for all $\mathbf{u} : \mathbb{R}_0^+ \to \mathbb{R}^p$ the following holds:*

$$x \succeq_x x',\ \mathbf{w} \succeq_w \mathbf{w}' \Rightarrow \forall t \geq 0,\ \Phi(t, x, \mathbf{u}, \mathbf{w}) \succeq_x \Phi(t, x', \mathbf{u}, \mathbf{w}').$$

If in addition the partial orderings $\succeq_x$ and $\succeq_w$ are induced by the positive orthants $\mathbb{R}_+^n$ and $\mathbb{R}_+^q$ respectively, (1.8) is cooperative.

The characterization of cooperative systems in Propositions 1.6 and 1.7 can similarly be adapted to (1.8). Although these results are applied to (1.8), we write them using the original vector field f of (1.1).

Proposition 1.11. *The system (1.8) with locally Lipschitz vector field f is cooperative if and only if the following implication holds for all $i \in \{1, \dots, n\}$:*

$$x \succeq_x x',\ x_i = x_i',\ w \succeq_w w' \Rightarrow \forall u \in \mathbb{R}^p,\ f_i(x, u, w) \geq f_i(x', u, w').$$

If the vector field f is continuously differentiable, (1.8) is cooperative if and only if for all $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$, $w \in \mathbb{R}^q$, $i, j \in \{1, \dots, n\}$, $j \neq i$, $k \in \{1, \dots, q\}$:

$$\frac{\partial f_i}{\partial x_j}(x, u, w) \geq 0,\ \frac{\partial f_i}{\partial w_k}(x, u, w) \geq 0.$$

1.2 Assumptions

In this section, we list the main assumptions that are used in the results presented in Chapters 2 to 4 to clarify the properties that can be expected from a system considered in this thesis. Note that not all these assumptions are required for all the results: for each result in the next chapters, the list of the required assumptions is clearly stated.

Even though this list of assumptions may seem fairly limiting the scope of application of our work, the assumptions presented below actually come from the generalization of preliminary results initially obtained on the application presented in Chapter 5. Our objective in Chapters 2 to 4 is thus to provide the widest class of systems where this initial work can be applied. This generalization resulted in introducing Assumptions 1, 1′, 2 and 3 below to be used in some of the results of the next chapters.

1.2.1 System description

We are interested in a system described by the differential equation

$$\dot{x} = f(x, u, w), \tag{1.1}$$

with state $x \in \mathbb{R}^n$, control input $u \in \mathbb{R}^p$ and disturbance input $w \in \mathbb{R}^q$. To use Propositions 1.6 and 1.9, we require the vector field f of (1.1) to be locally Lipschitz in its continuous variables. This assumption is considered to be always verified and will not be repeated. When stronger assumptions on the vector field are necessary, such as continuous differentiability to use its partial derivatives or Proposition 1.7,

they are specified in the result statement. For completeness of this section, let us remind the remaining notations of the system description. The trajectories of (1.1) from initial state x_0 are denoted as $\Phi(\cdot, x_0, \mathbf{u}, \mathbf{w})$ where $\mathbf{u}$ and $\mathbf{w}$ are functions of time. Under state-feedback $\mathsf{u} : \mathbb{R}^n \to \mathbb{R}^p$, we similarly denote the closed loop system as

$$\dot{x} = f_{\mathsf{u}}(x, w) = f(x, \mathsf{u}(x), w), \tag{1.2}$$

and its trajectories as $\Phi_{\mathsf{u}}(\cdot, x_0, \mathbf{w})$.

1.2.2 Monotonicity

The monotonicity property is the basis of all results presented in this thesis. More precisely, we focus on cooperative systems as in Definition 1.5, which means that the partial orderings involved in the monotonicity definition are the classical componentwise inequalities.

Assumption 1. *System (1.1) is cooperative as in Definition 1.5 with bounded inputs:* $u \in [\underline{u}, \overline{u}]$ *and* $w \in [\underline{w}, \overline{w}]$.

Combining the assumption of monotonicity with bounded inputs is crucial for robustness analysis. Indeed, if we consider Definition 1.5 and the previous illustrative examples in Figures 1.1 and 1.2, we can see that with an initial state $x'' \in [x', x]$ and input functions $\mathbf{u}$, $\mathbf{u}'$, $\mathbf{u}''$ and $\mathbf{w}$, $\mathbf{w}'$, $\mathbf{w}''$ such that $\mathbf{u}''(t) \in [\mathbf{u}'(t), \mathbf{u}(t)]$ and $\mathbf{w}''(t) \in [\mathbf{w}'(t), \mathbf{w}(t)]$ for all $t \in \mathbb{R}_0^+$, we obtain $\Phi(t, x'', \mathbf{u}'', \mathbf{w}'') \in [\Phi(t, x', \mathbf{u}', \mathbf{w}'), \Phi(t, x, \mathbf{u}, \mathbf{w})]$, for all $t \in \mathbb{R}_0^+$. Hence, having bounded inputs allows considering only their extremal values since we know that all other behaviors of the system are necessarily bounded by the extremal behaviors. Even though the cooperativeness is required throughout this thesis, it is not always needed with respect to all variables of (1.1). This is why we introduce a second version of this assumption.

Assumption 1′. *System (1.8) is cooperative as in Definition 1.10 with bounded inputs:* $u \in [\underline{u}, \overline{u}]$ *and* $w \in [\underline{w}, \overline{w}]$.

Assumption 1 naturally implies Assumption 1′. These two assumptions are required in the following cases:

- Assumption 1 is used in Chapters 2 and 4, where we need (1.1) to be cooperative with respect to all its variables as in Definition 1.5,
- Assumption 1′ is used in Chapter 3, where we do not need cooperativeness with respect to the control input u and then use Definition 1.10 on the time-dependent vector field $F(t, x, w) = f(x, u(t), w)$ from (1.8).

1.2.3 Local control

We say that the system satisfies the local control property if any component of the control input directly influence a single component of the state.

Definition 1.12 (Local control)**.** *System (1.1) satisfies the local control property if it can be written as follows:*

$$\dot{x}_i = f_i(x, u_i, w),\ \forall i \in \{1, \dots, n\}, \tag{1.9}$$

where f_i is the i^{th} component of the vector field, u_i represents the control inputs with a direct influence on the state x_i and u_i and u_j are disjoint for all $j \neq i$.

Note that having each control input influencing a single state does not mean that each state is directly influenced by exactly one control input: there may be several, one or none and as a result u_i may be a vector, a scalar or the empty set. Hence, as illustrated in the example below, Assumption 2 is not as restrictive as it seems.

Example 1.1. For linear systems $\dot{x} = Ax + Bu$, the local control property from Definition 1.12 is satisfied if and only if each column of the matrix B has exactly a single non-zero element. With more than one non-zero element in the i^{th} column, u_i influences two state variables. With less than one, the whole system is independent of u_i, which thus is not an input of the system. As stated above, there can be any number of non-zero elements per rows of the matrix B.

For example, the system $y^{(3)} = u + v$ can be shown to satisfy Definition 1.12 when we write it in state space form:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix},\ y = x_1. \qquad \triangle$$

Assumption 2. *System (1.1) satisfies the local control property.*

In Chapter 2, this assumption is used for most results based on robust controlled invariance since in some situations the control input needs to steer two state components in opposite directions (e.g. increase x_i and decrease x_j), which cannot be done with a control input influencing both states in a positive way (from the cooperativeness of the system). For this reason, only the initial result on robust invariance in Chapter 2 does not need Assumption 2 since it considers the control input u as a disturbance. In Chapters 3 and 4, Assumption 2 is not required although it could be useful in Chapter 4 to facilitate the decomposition in the compositional method.

Remark 1.13. *Under Assumptions 1 and 2 and with a decentralized state-feedback* u *(meaning that* $\mathsf{u}_i(x) = \mathsf{u}_i(x_i)$ *for all* i*), the closed-loop system (1.2) is cooperative.*

This is particularly easy to see when we can use Proposition 1.7 since with the assumptions from Remark 1.13,

$$f_{\mathsf{u},i}(x, w) = f_i(x, \mathsf{u}(x), w) = f_i(x, \mathsf{u}_i(x_i), w)$$

and the partial derivatives of $f_{\mathsf{u},i}$ with respect to w and x_j with $j \neq i$ are the same as the partial derivatives of f.

1.2.4 Static input-state characteristic

We extend the definition from [AS03] to system (1.1) with both control and disturbance inputs.

Definition 1.14. *System (1.1) has a static input-state characteristic denoted as* $k_x : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^n$ *if for each pair* (u, w) *of constant inputs, (1.1) has a unique globally asymptotically stable equilibrium* $k_x(u, w)$*.*

Assumption 3. *System (1.1) has a static input-state characteristic.*

This assumption is relatively less important than the others as all the main results can still be applied if it is not satisfied. It is only useful when we need to use the unique equilibrium corresponding to a given pair of constant inputs. This is the case in Section 2.2 for the result on robust invariance, which is useful to restrict the analysis of the system to a smaller domain, but it is not required for the subsequent developments. The second use of Assumption 3 is in one example to create a support function for the robust set stabilization in Section 2.5.

Remark 1.15. *Under Assumptions 1 and 3, the static input-state map is also monotone [AS03]:*

$$u \succeq_u u',\ w \succeq_w w' \Rightarrow k_x(u, w) \succeq_x k_x(u', w').$$

This is easily proven by applying the monotonicity definition 1.5 with identical initial state and taking the limit when $t \to +\infty$.

1.3 Illustration example

Before the more involved application to the temperature regulation in an experimental smart building in Chapter 5, we want to illustrate the main concepts from Chapters 2 to 4 through simpler examples. In this section, we thus introduce two such systems and show that they satisfy all the assumptions from Section 1.2.

1.3.1 Discrete diffusion equation

In the simplest example, we consider the temperature diffusion in a rod. We assume that the temperature is uniform in any cross-section of the rod, which can thus be approximated by a one-dimensional object whose state only varies along its length. We consider a version of the diffusion equation that is continuous in time and discretized in space: the rod is partitioned into several segments, each assumed to have a uniform temperature. The system of interest is sketched in Figure 1.4 with a rod partitioned into four segments where the temperature of the leftmost and rightmost segments are respectively set by a control input $u \in [\underline{u}, \overline{u}] \subseteq \mathbb{R}$ and a disturbance input $w \in [\underline{w}, \overline{w}] \subseteq \mathbb{R}$. The state $x \in \mathbb{R}^2$ of the system corresponds to the temperature of both central segments. Although this system could easily be extended to a finer partition of the rod, for easier planar visualization of the state space we only consider the case described by Figure 1.4 with two state variables.

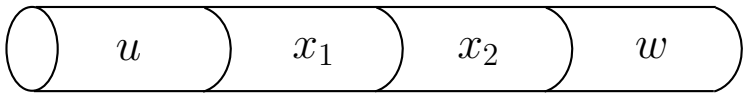


Figure 1.4 – Rod partitioned into four segments with the temperature of both extremities set by the control and disturbance inputs u and w, respectively.

The dynamics of this system can be written as in (1.1) by the following linear differential equation:

$$\dot{x} = f(x,u,w) = \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix} x + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u + \begin{pmatrix} 0 \\ 1 \end{pmatrix} w \tag{1.10}$$

Proposition 1.16. *System (1.10) satisfies Assumptions 1, 2 and 3.*

Proof. Since the vector field f of (1.10) is continuously differentiable, we can prove that (1.10) is cooperative using Proposition 1.7 as all the following partial derivatives are non-negative:

$$\frac{\partial f_1}{\partial x_2} = 1,\ \frac{\partial f_1}{\partial u} = 1,\ \frac{\partial f_1}{\partial w} = 0, \qquad \frac{\partial f_2}{\partial x_1} = 1,\ \frac{\partial f_2}{\partial u} = 0,\ \frac{\partial f_2}{\partial w} = 1.$$

In addition, both scalar inputs u and w can reasonably be assumed to be bounded in $[\underline{u}, \overline{u}] \subseteq \mathbb{R}$ and $[\underline{w}, \overline{w}] \subseteq \mathbb{R}$ respectively, which then implies that Assumption 1 holds.

The local control property from Definition 1.12 and Assumption 2 is also immediately satisfied since the control input u affects only the single state variable x_1.

Let A, B_u and B_w be the matrices such that (1.10) can be written as $\dot{x} = Ax + B_u u + B_w w$. Since the eigenvalues of A are -3 and -1, the system is stable and the unique globally asymptotically stable equilibrium corresponding to a pair of constant inputs (u, w) is given by the static input-state characteristic: $k_x(u,w) = -A^{-1}(B_u u + B_w w)$. We finally obtain:

$$k_x(u,w) = \frac{1}{3}\begin{pmatrix} 2u + w \\ u + 2w \end{pmatrix}. \tag{1.11}$$

□

For all the examples in Chapters 2 to 4 referring to this system, we consider the following input intervals:

$$u \in [18, 30], \qquad w \in [15, 21].$$

1.3.2 Coupled tanks

While the temperature diffusion model introduced in the previous section is particularly simple and can be useful to illustrate basic concepts, we want to show that the results in Chapters 2 to 4 can also be applied to more complex and realistic systems.

The non-linear system considered in this section is inspired from the coupled-tank experiment described in [ALA]. This system, sketched in Figure 1.5, consists in two identical water tanks of height 30 cm, cross-sectional area $A = 4.425\,\mathrm{cm}^2$ and with an orifice of cross-sectional area $a = 0.476\,\mathrm{cm}^2$. The outflow from tank 1 goes into tank 2 and the outflow from tank 2 goes into a basin. The water levels in tank 1 and 2 are denoted as x_1 and x_2, respectively, and correspond to the state variables. Two pumps of constants $K_1 = 4.6\,\mathrm{cm}^3/\mathrm{V/s}$ and $K_2 = 2\,\mathrm{cm}^3/\mathrm{V/s}$ supply water to the tanks 1 and 2, respectively. These pumps are controlled in voltage with $u_1, u_2 \in [0, 22]\,\mathrm{V}$. Finally, there is a possible leak at the bottom of tank 2. The corresponding outflow w can take values between $\underline{w} = -20\,\mathrm{cm}^3/\mathrm{s}$ (maximal leak) and $\overline{w} = 0$ (no leak).

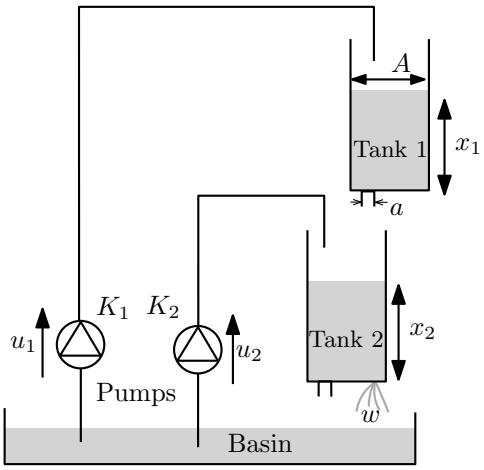


Figure 1.5 – Coupled-tank system with two pumps and a leak on tank 2.

Applying Bernoulli's principle for flows through small orifices, the outflow from tank i is given by $a\sqrt{2gx_i}$, where $g = 980\,\mathrm{cm/s}^2$ is the gravitational constant. The non-linear model of the variations of the water level in each tank thus has the following dynamics:

$$\begin{pmatrix}\dot{x}_1\\ \dot{x}_2\end{pmatrix} = \frac{a\sqrt{2g}}{A}\begin{pmatrix}-1 & 0\\ 1 & -1\end{pmatrix}\begin{pmatrix}\sqrt{x_1}\\ \sqrt{x_2}\end{pmatrix} + \frac{1}{A}\begin{pmatrix}K_1 & 0\\ 0 & K_2\end{pmatrix}\begin{pmatrix}u_1\\ u_2\end{pmatrix} + \frac{1}{A}\begin{pmatrix}0\\ 1\end{pmatrix} w, \quad (1.12)$$

that we write as $\dot{x} = f(x, u, w)$ or as $\dot{x} = A\sqrt{x} + B_u u + B_w w$ where the square root is taken componentwise. To avoid the origin point where (1.12) is not locally Lipschitz, we prove the assumptions for strictly positive water levels and we will later only consider control objectives where no tank is empty.

Proposition 1.17. *System (1.12) satisfies Assumptions 1 and 2.*

Proof. We have already provided the bounds for both the control and disturbance inputs. We then prove the cooperativeness using the partial derivatives of the vector

field f for $x_1 > 0$ and $x_2 > 0$:

$$\frac{\partial f_1}{\partial x_2} = 0,\ \frac{\partial f_1}{\partial u_1} = \frac{K_1}{A} > 0,\ \frac{\partial f_1}{\partial u_2} = \frac{\partial f_1}{\partial w} = 0,$$
$$\frac{\partial f_2}{\partial x_1} = \frac{a}{2A}\sqrt{\frac{2g}{x_1}} \geq 0,\ \frac{\partial f_2}{\partial u_1} = 0,\ \frac{\partial f_2}{\partial u_2} = \frac{K_2}{A} > 0,\ \frac{\partial f_2}{\partial w} = \frac{1}{A} > 0.$$

Therefore, the system is cooperative and Assumption 1 is satisfied. Note that the disturbance w was chosen with negative values to ensure $\partial f_2/\partial w > 0$: increasing w (thus reducing the leak) has a positive effect on the water level x_2.

The local control property from Assumption 2 is also satisfied since the control matrix $B_u = \begin{pmatrix} K_1/A & 0 \\ 0 & K_2/A \end{pmatrix}$ has a single non-zero value per column: u_1 and u_2 only directly influence x_1 and x_2, respectively.

The static input-state characteristic from Assumption 3 is not used on the examples on this system. □

The main goal of this system is to control the water level of tank 2 in a range specified in each example. There is no restriction on the water level of tank 1 apart from the fact that it does not overflow ($x_1 \leq 30\,\text{cm}$). Since the cooperativeness of the system is only proven for strictly positive water levels, we also impose $x_1 \geq 1\,\text{cm}$. In Chapters 3 and 4, in addition to these safety specification we penalize the use of the pumps, with a bigger weight on u_2: ideally, we want to control the level x_2 only with u_1, but the second pump is provided as a back-up to be used only when the specifications cannot be realized or when we do not have information on the voltage applied to the first pump (compositional approach).

Chapter 2

Robust controlled invariance

In this chapter, we study robust controlled invariance of a set and in particular of an interval as defined in (1.3). This notion describes the ability to control a system such that its state is maintained in a set at all time and for any value of the disturbance, assuming that this set contains the initial state of the system.

We first motivate the usage of a robust controlled invariant interval and review some of the related literature in Section 2.1. Section 2.2 presents some preliminary results on a related notion without control while the definition and main results on robust controlled invariance are given in Section 2.3. The robust local stabilizability in Section 2.4 is closely related to the characterization of a robust controlled invariant interval reduced to a single point. Finally, in Section 2.5, we extend the invariance problem to a stabilization problem where the goal is to reach and stay in an interval when the initial state possibly lies outside of this interval. The main results are illustrated on the discrete diffusion equation and coupled-tank system presented in Section 1.3.

The results in Sections 2.2 to 2.4 were first published in [MGW13]. Experimental implementations of the robust controlled invariance for the temperature control on the system described in Chapter 5 then appeared in [MNGW13, MNGW14]. A more in-depth description of the robust controlled invariance (Section 2.3) and the introduction of the robust set stabilization (Section 2.5) are given in [MGWa], with an experimental implementation on the same application.

2.1 Motivations and related work

Stability and invariance When dealing with systems subject to disturbances, the stability of the controlled system can be approached in several ways depending on the type of disturbance and the control objectives. If the disturbances are measured, we can consider classical Lyapunov stability [LSL61] and apply a controller that not only depends on the state feedback, but also adapts to the measured values of the disturbances. On the other hand, when the disturbances are unknown, we need to rely on *robust control* approaches that define alternative stability notions. If we have no other information on the disturbance, we can study the input to state stability, stating that the undisturbed system is stable and a bounded disturbance implies a bounded variation of the state from its equilibrium [Son08]. With known bounds

on the disturbances, the robustness can be approached with the notion of practical stability, introduced in [LSL61]. This notion is the extension of the classical stability to a set: the disturbances prevent the stability in a particular state but their bounded values allow to control the system in a set around this state.

The notion of practical stability is our main motivation to consider the control in a set rather than in a state. In addition, since we may also want to control the system in larger sets, we discard the notion of (practical) stability and directly work with an invariance control objective: we want to maintain the state of the system in a set of the state space [Bla99]. This is motivated by the fact that even practical stability cannot be realized in some systems when the range of disturbance values is too large (see Example 2.4 in Section 2.4).

Another motivation to consider invariance instead of stability comes from our application of temperature control in buildings presented in Chapter 5, which was the starting point of the work presented in this thesis. Due to the discretized control input (possibly just on/off) in the symbolic approach of Chapters 3 and 4, this situation recalls the classical thermostat example used in any lecture introducing non-linear control. In this example, an on/off thermostat associated with a temperature setpoint may lead to infinitely fast switching and damage the actuator. To prevent this phenomenon, we need to change the control strategy into a hysteresis as in Figure 2.1 by splitting the switching triggers T_{on} and T_{off}. This is equivalent to widening the specifications from a setpoint to an interval.

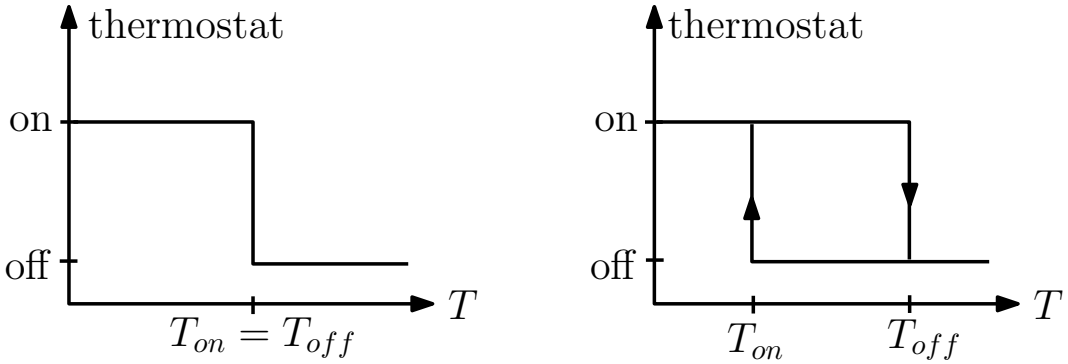


Figure 2.1 – On/off (left) and hysteresis (right) thermostat controller.

Invariance An extensive survey on the topic of invariance and its applications in control has been done by Blanchini [Bla99]. In this paper, the notion of positive invariance is described as the property that trajectories initialized in a set remain in this set forever.

Definition 2.1 (Positive invariance)**.** *For the autonomous system $\dot{x} = f(x)$ with trajectories Φ, the set $\mathcal{S}$ is said to be positively invariant if the following implication holds:*

$$x_0 \in \mathcal{S} \Rightarrow \forall t \geq 0,\ \Phi(t, x_0) \in \mathcal{S}.$$

Similarly, Blanchini defines an invariant set as a set satisfying the implication of Definition 2.1 for both positive and negative times ($x_0 \in \mathcal{S} \Rightarrow \forall t \in \mathbb{R},\ \Phi(t, x_0) \in \mathcal{S}$).

Since we consider physical systems, we have no interest in the notion of negative invariance (backward in time). Thus, when referring to *invariance* in what follows, we actually consider *positive invariance* as in Definition 2.1. When a control input is used to enforce the invariance of the system in a set, we talk about *controlled invariance*, independently introduced in [BM69] and [WM70], or about viability [Aub91]. Some of the main results on controlled invariant sets for linear systems are given in [TSH01]. With the addition of disturbances influencing the dynamics of the system, we are interested in *robust controlled invariant* sets (or simply robust invariant if there is no control input). Similarly to Definition 2.1, a set is robust invariant if it is invariant for any value of the disturbances, and it is robust controlled invariant when there exists a control strategy such that the controlled system is robust invariant. In the viability theory, a notion similar to robust controlled invariant sets is that of *discriminating domains*, where the disturbance corresponds to the plays of the environment in a game against the system [Aub91].

Invariance and monotonicity For the class of systems (1.1) considered in this work and introduced in Chapter 1, no assumption is made on their linearity. Hence the results on linear systems from [TSH01] cannot be applied. On the other hand, Assumption 1 combined with the choice of an interval (1.3) as our goal for invariance greatly simplifies both notions of invariance and robustness in the characterization of robust (controlled) invariant sets (Sections 2.2 and 2.3). Indeed, when all the variables (state and inputs) of a cooperative system are in sets with both a lower and an upper bound, the classical robustness criterion of considering the worst cases is straightforward since the extremal behaviors of the system are obtained with the extremal values of its state and inputs.

There is relatively few works on (controlled) invariance of non-linear monotone systems. Sufficient conditions for invariance of an autonomous system in an interval are presented in [ATS09] for the class of monotone multi-affine systems. Methods to obtain upper and lower approximations of the maximal controlled invariant set of a monotone discrete-time system without disturbance are considered in [LDGR07]. A reasoning close to the one on robust invariance in Section 2.2 is carried out in [RMC10] and [RMC09] for uncertain monotone and mixed-monotone systems respectively, where they bound the behavior of the uncertain system by those of two dynamical systems which do not depend on the uncertainty. They further extend these results to piecewise (mixed-) monotone systems by creating a hybrid automaton where the previous results can be applied to each state. Finally, the authors of [GDV14] are interested in robust controlled invariance for input-output order-preserving systems (a super-class of monotone systems) where the robustness analysis is concerned with imperfect state information.

2.2 Robust invariance

As a first step toward the definition of the main notion of robust controlled invariant set in Section 2.3, we introduce the simpler notion of robust invariance. In Definition 2.1, for an autonomous system $\dot{x} = f(x)$, an invariant set is described as a set that contains all trajectories $\Phi(\cdot, x_0)$ of the system as long as the initial state x_0 is

in this set. For system (1.1) with both control and disturbance inputs restricted to the intervals in Assumption 1, we can add to this definition a notion of robustness with respect to the inputs when we want an invariant set common to all possible input functions.

Definition 2.2 (Robust invariance)**.** *A set $\mathcal{S}$ is robust invariant for (1.1) if,*

$$\forall x_0 \in \mathcal{S},\ \forall \mathbf{u} \in [\underline{u}, \overline{u}],\ \forall \mathbf{w} \in [\underline{w}, \overline{w}],\ \forall t \geq 0,\ \Phi(t, x_0, \mathbf{u}, \mathbf{w}) \in \mathcal{S}.$$

It can easily be seen from Definition 2.2 that any robust invariant set contains all reachable states when the system is initialized in this set. However, the converse is not true since a robust invariant set may contain states that are not reachable from other states in the set. To minimize the quantity of such unreachable states in a robust invariant set, it is thus natural to look for the smallest over-approximation of the reachable set expressed with robust invariance. To exploit the advantages of the monotonicity property as in Section 1.2.2, we focus on robust invariant intervals $[\underline{x}, \overline{x}]$ and give a characterization of the minimal robust invariant interval, where minimality refers to the set inclusion.

Theorem 2.3. *Under Assumption 1, $[\underline{x}, \overline{x}]$ is robust invariant if and only if*

$$\begin{cases} f(\overline{x}, \overline{u}, \overline{w}) \preceq_x 0, \\ f(\underline{x}, \underline{u}, \underline{w}) \succeq_x 0. \end{cases}$$

In addition, if Assumption 3 holds, then the minimal robust invariant interval is $[k_x(\underline{u}, \underline{w}), k_x(\overline{u}, \overline{w})]$.

Proof. Since the state of the system varies continuously, we can say that $[\underline{x}, \overline{x}]$ is robust invariant if and only if for any element x of the boundary of $[\underline{x}, \overline{x}]$ the flow $\Phi(t, x, \mathbf{u}, \mathbf{w})$ does not leave the interval. This is equivalent to having the vector field at x oriented toward the interior of the interval for all $u \in [\underline{u}, \overline{u}]$ and $w \in [\underline{w}, \overline{w}]$. Thus it is clear that the conditions in Theorem 2.3 are necessary. Let us show that they are also sufficient under Assumption 1. From the Kamke-Müller condition in Proposition 1.6, we have for all $x \in [\underline{x}, \overline{x}]$, $u \in [\underline{u}, \overline{u}]$, $w \in [\underline{w}, \overline{w}]$ and $i \in \{1, \ldots, n\}$,

$$\begin{cases} x_i = \overline{x}_i \Rightarrow f_i(x, u, w) \leq f_i(\overline{x}, \overline{u}, \overline{w}) \leq 0, \\ x_i = \underline{x}_i \Rightarrow f_i(x, u, w) \geq f_i(\underline{x}, \underline{u}, \underline{w}) \geq 0. \end{cases}$$

Therefore $[\underline{x}, \overline{x}]$ is robust invariant since the vector field always points toward the interior of the interval when the state is on its boundary.

Now, assume that Assumption 3 holds. By definition of the equilibrium points $k_x(\overline{u}, \overline{w})$ and $k_x(\underline{u}, \underline{w})$, we have

$$\begin{cases} f(k_x(\overline{u}, \overline{w}), \overline{u}, \overline{w}) = 0, \\ f(k_x(\underline{u}, \underline{w}), \underline{u}, \underline{w}) = 0. \end{cases}$$

From our previous analysis, $[k_x(\underline{u}, \underline{w}), k_x(\overline{u}, \overline{w})]$ is robust invariant. Also, any robust invariant interval would contain $k_x(\underline{u}, \underline{w})$ and $k_x(\overline{u}, \overline{w})$ as these are globally asymptotically stable equilibria for constant inputs $\mathbf{u} = \underline{u}$, $\mathbf{w} = \underline{w}$ and $\mathbf{u} = \overline{u}$, $\mathbf{w} = \overline{w}$, respectively. Hence, the robust invariant interval $[k_x(\underline{u}, \underline{w}), k_x(\overline{u}, \overline{w})]$ is minimal with respect to set inclusion. □

From Definition 2.2, we know that the minimal robust invariant interval is the smallest interval over-approximation of the reachable set if the state is initialized in this interval. It thus can be useful in subsequent studies and numerical implementations to restrict the analysis of system (1.1) to that region. Note that in the absence of Assumption 3, there may not exist a minimal robust invariant interval.

Example 2.1. Applying Theorem 2.3 to the temperature diffusion example (1.10) presented in Section 1.3.1 with $u \in [18, 30]$ and $w \in [15, 21]$, we have that the interval

$$[k_x(\underline{u}, \underline{w}), k_x(\overline{u}, \overline{w})] = \left[\begin{pmatrix} 17 \\ 16 \end{pmatrix}, \begin{pmatrix} 27 \\ 24 \end{pmatrix} \right]$$

is the minimal robust invariant interval. This implies that for any input functions $\mathbf{u}$ and $\mathbf{w}$, the state of (1.10) always stays in this interval if it started there. In addition, we know that any other robust invariant intervals $[\underline{x}, \overline{x}]$ are such that $[k_x(\underline{u}, \underline{w}), k_x(\overline{u}, \overline{w})] \subseteq [\underline{x}, \overline{x}]$. △

2.3 Robust controlled invariance

For the robust controlled invariance, we extend Definition 2.2 by keeping the robustness considerations only on the disturbance input w and taking advantage of the control input u to actively counteract the effects of the disturbance. We thus define a robust controlled invariant set for system (1.1) as a robust invariant set for the closed-loop (1.2) obtained from the use of an invariance feedback controller. We use the notation Φ_{u} of the trajectories of the closed-loop with feedback $\mathsf{u} : \mathbb{R}^n \to \mathbb{R}^p$.

Definition 2.4 (Robust controlled invariance)**.** *A set $\mathcal{S}$ is robust controlled invariant if there exists a controller $\mathsf{u} : \mathcal{S} \to [\underline{u}, \overline{u}]$ such that*

$$\forall x_0 \in \mathcal{S},\ \forall \mathbf{w} \in [\underline{w}, \overline{w}],\ \forall t \geq 0,\ \Phi_{\mathsf{u}}(t, x_0, \mathbf{w}) \in \mathcal{S}.$$

We call u an invariance controller in $\mathcal{S}$.

Note that with the use of an invariance controller as in Definition 2.4, we can greatly reduce the size of the robust invariant sets for the closed-loop system compared to those for system (1.1) obtained in Definition 2.2 with robustness considerations on the control input u. Using the monotonicity property, we obtain a characterization of robust controlled invariant intervals expressed only in terms of the vector field and using the extremal values of the state x and inputs u and w.

Theorem 2.5. *Under Assumptions 1 and 2, the interval $[\underline{x}, \overline{x}]$ is robust controlled invariant if and only if*

$$\begin{cases} f(\overline{x}, \underline{u}, \overline{w}) \preceq_x 0, \\ f(\underline{x}, \overline{u}, \underline{w}) \succeq_x 0. \end{cases}$$

Proof. We prove necessity by contrapositive. Assume that $f(\overline{x}, \underline{u}, \overline{w}) \not\preceq_x 0$. This means that there exists $i \in \{1, \ldots, n\}$ such that $f_i(\overline{x}, \underline{u}, \overline{w}) > 0$. With Proposition 1.6, it follows that $\forall u \in [\underline{u}, \overline{u}]$, $f_i(\overline{x}, u, \overline{w}) \geq f_i(\overline{x}, \underline{u}, \overline{w}) > 0$. Thus no value of

the control input u can make the vector field at $\overline{x}$ point toward the interior of the interval, making it non-invariant. We can have a similar reasoning if there exists $i \in \{1, \dots, n\}$ such that $f_i(\underline{x}, \overline{u}, \underline{w}) < 0$.

Let us now prove sufficiency. By Assumption 2, we have that for all $i \in \{1, \dots, n\}$, $f_i(x, u, w) = f_i(x, u_i, w)$ where the inputs $u_i \in [\underline{u}_i, \overline{u}_i]$ only have a direct influence on the i^{th} component of the vector field. Then, by Proposition 1.6, we have that for all $x \in [\underline{x}, \overline{x}]$, $w \in [\underline{w}, \overline{w}]$ and $i \in \{1, \dots, n\}$,

$$\begin{cases} x_i = \overline{x}_i \Rightarrow f_i(x, \underline{u}_i, w) \leq f_i(\overline{x}, \underline{u}_i, \overline{w}) \leq 0, \\ x_i = \underline{x}_i \Rightarrow f_i(x, \overline{u}_i, w) \geq f_i(\underline{x}, \overline{u}_i, \underline{w}) \geq 0. \end{cases}$$

Since the vectors u_i and u_j are independent for $i \neq j$, it follows from the previous inequalities, that for any state x on the boundary of the interval $[\underline{x}, \overline{x}]$ there exists a value of the control input $\mathsf{u}(x) \in [\underline{u}, \overline{u}]$ such that the vector field at x points toward the interior of the interval for any value of the disturbance. Using such controller u, we can always force the flow toward the interior when the state reaches the boundary of the interval. This implies the robust controlled invariance of the interval. □

The first condition of Theorem 2.5 states that when the current state is on the upper bound of the interval with the maximal value of the disturbance, the minimal value of the control can force the state to be non-increasing. Similarly for the second condition, the maximal control can force a non-negative vector field when the state is on the lower bound of the interval with the minimal value of the disturbance. Thus, an interpretation of Theorem 2.5 is that if the extremal values of the control input can maintain the vector field pointing inside the interval in the worst conditions, then the invariance in the interval is satisfied for any other condition.

Example 2.2. For the two-dimensional temperature diffusion system (1.10) from Section 1.3.1, Theorem 2.5 implies the following. An interval $[\underline{x}, \overline{x}] \in \mathbb{R}^2$ is robust controlled invariant if and only if it satisfies:

$$\begin{cases} -2\underline{x_1} + \underline{x_2} + \overline{u} \geq 0 \\ \underline{x_1} - 2\underline{x_2} + \underline{w} \geq 0 \end{cases} \quad \text{and} \quad \begin{cases} -2\overline{x_1} + \overline{x_2} + \underline{u} \leq 0 \\ \overline{x_1} - 2\overline{x_2} + \overline{w} \leq 0 \end{cases} \tag{2.1}$$

with $[\underline{u}, \overline{u}] = [18, 30]$ and $[\underline{w}, \overline{w}] = [15, 21]$. These conditions are displayed in Figure 2.2. First, note that the black dashed interval corresponds to the minimal robust invariant interval computed in Example 2.1. The robust controlled invariance equations (2.1) presents two conditions on each bound $\underline{x}$ and $\overline{x}$ of the interval. The intersection of the conditions on $\underline{x}$ is the blue set in Figure 2.2 and the limits of the corresponding inequalities are the blue lines. Similarly, the red set and red lines represent the intersection and the limits of the conditions on $\overline{x}$. Thus, according to Theorem 2.5, an interval is robust controlled invariant if and only if its lower bound lies in the blue set and its upper bound is in the red set. We can notice on Figure 2.2 that such an interval usually needs to be wider on its second state component x_2 than on x_1. This is due to the fact that in system (1.10), the control input u only affects directly the state x_1 and then the conditions on $[\underline{x_2}, \overline{x_2}]$ are closer to the robust invariance.

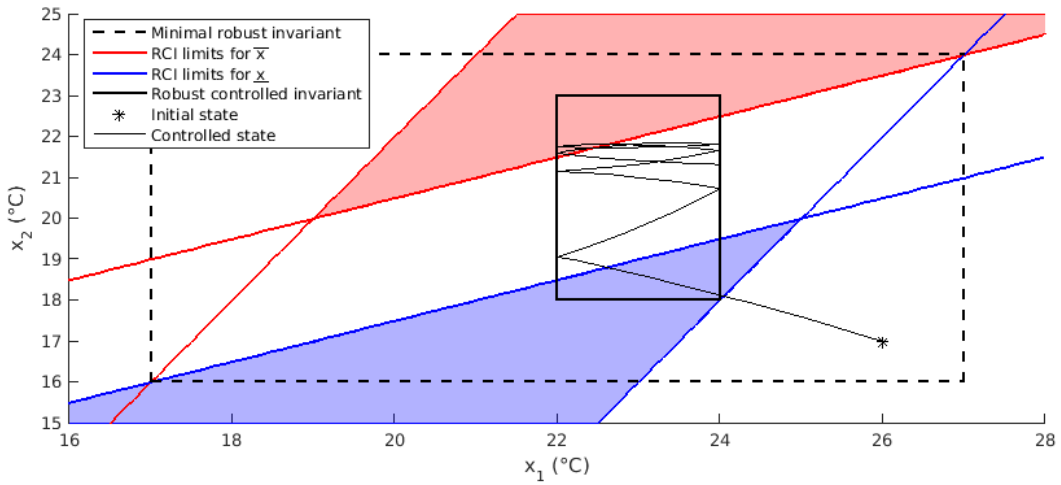


Figure 2.2 – Example 2.2: possible choices for a robust controlled invariant interval and bang-bang control.

We also include in Figure 2.2 a control application. We first choose the black robust controlled invariant interval in Figure 2.2 with its lower bound $\underline{x} = (22; 18)$ in the blue set and its upper bound $\overline{x} = (24; 23)$ in the red set. As in the comment following Theorem 2.5, we want to show that we can maintain the state in the interval solely by using the extremal values of the control input when the state reaches the boundary of $[\underline{x}, \overline{x}]$. We thus consider the family of controllers defined as follows:

$$\mathsf{u}(x) \begin{cases} = \underline{u}, & \text{if } x_1 = \overline{x_1}, \\ \in [\underline{u}, \overline{u}], & \text{if } x_1 \in (\underline{x_1}, \overline{x_1}), \\ = \overline{u}, & \text{if } x_1 = \underline{x_1}. \end{cases} \tag{2.2}$$

The second condition of (2.2) means that we can use any value of the controller when x_1 is in the interior of the interval. In our case, we apply a bang-bang control where we simply keep the previous value of u. The simulation is initialized in $x_0 = (23; 19.5)$ and the disturbance w is set as a sine wave between $\underline{w}$ and $\overline{w}$. As predicted, the controlled state, drawn in black in Figure 2.2, bounces between $\underline{x_1}$ and $\overline{x_1}$ and never leaves the interval. △

In the above $2D$ example, we have seen that if we replace the inequalities in Theorem 2.5 by equalities, we define 4 curves (2 for $\underline{x}$ and 2 for $\overline{x}$) giving the boundaries of the blue and red sets. For a generalization to a n-dimensional system, taking equalities in Theorem 2.5 defines n manifolds of the state space, each of dimension $n-1$, for the boundary of the set where to choose $\underline{x}$ and n others for the upper bound $\overline{x}$.

Note that without the local control property from Assumption 2, the conditions in Theorem 2.5 are still necessary but not sufficient. We illustrate this case in the following example.

Example 2.3. Consider the system of $\mathbb{R}^2$ defined as follows:

$$\dot{x} = f(x,u,w) = \begin{pmatrix} u + w_1 \\ u + w_2 \end{pmatrix},$$

where $u \in [-1,1]$, $w \in [-\varepsilon,\varepsilon]^2$ and $\varepsilon < 1$. This system is clearly cooperative and satisfies the conditions from Theorem 2.5 for any interval $[\underline{x},\overline{x}]$:

$$f(\overline{x},\underline{u},\overline{w}) = \begin{pmatrix} -1+\varepsilon \\ -1+\varepsilon \end{pmatrix} \leq 0 \quad ; \quad f(\underline{x},\overline{u},\underline{w}) = \begin{pmatrix} 1-\varepsilon \\ 1-\varepsilon \end{pmatrix} \geq 0.$$

However, we can show that the interval is not robust controlled invariant as Assumption 2 is not satisfied. Consider the case where the state is on the bottom right vertex of the interval and the disturbance is such that $w = (\varepsilon,-\varepsilon)$, then $f((\overline{x_1},\underline{x_2}),u,(\varepsilon,-\varepsilon)) = \begin{pmatrix} u+\varepsilon \\ u-\varepsilon \end{pmatrix}$. To ensure the robust controlled invariance, it is necessary that in this situation we can find a control input u such that $f_1 \leq 0$ and $f_2 \geq 0$. As shown in Figure 2.3, this is not possible since for any value of the control input $u \in [-1,1]$ the vector field on the bottom right vertex points outside the interval. △

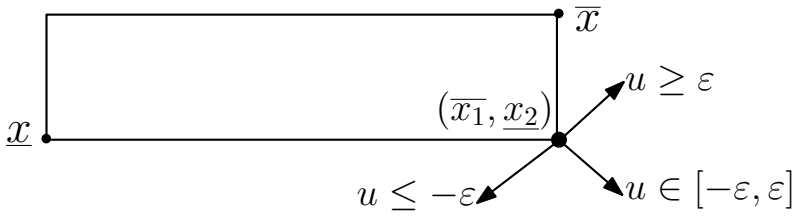


Figure 2.3 – Illustration of Example 2.3: Theorem 2.5 without Assumption 2.

In the proof of Theorem 2.5 and Example 2.2, we have shown that when an interval is robust controlled invariant, an invariance controller can be created by using only the extremal values of the control components u_i in the right situation. However, this is not necessary and we can give a characterization of the invariance controllers as follows.

Proposition 2.6. *Under Assumptions 1 and 2, let $[\underline{x},\overline{x}]$ be a robust controlled invariant. A controller $\mathsf{u} : [\underline{x},\overline{x}] \to [\underline{u},\overline{u}]$ is an invariance controller in $[\underline{x},\overline{x}]$ if and only if for all $i \in \{1,\ldots,n\}$:*

$$\mathsf{u}_i(x) \in \begin{cases} \underline{U}_i(x) = \{u_i \in [\underline{u_i},\overline{u_i}] |\ f_i(x,u_i,\overline{w}) \leq 0\} & \text{if } x_i = \overline{x_i}, \\ [\underline{u_i},\overline{u_i}] & \text{if } x_i \in (\underline{x_i},\overline{x_i}), \\ \overline{U}_i(x) = \{u_i \in [\underline{u_i},\overline{u_i}] |\ f_i(x,u_i,\underline{w}) \geq 0\} & \text{if } x_i = \underline{x_i}. \end{cases} \tag{2.3}$$

Proof. It is necessary and sufficient that for all x on the boundary of the interval $[\underline{x},\overline{x}]$, the vector field of the closed-loop system (1.2) at x points inside the interval

for all values of the disturbance. From Assumption 2, this is the case if and only if for all $w \in [\underline{w}, \overline{w}]$ we have $f_i(x, \mathsf{u}_i(x), w) \leq 0$ whenever a state component x_i reaches $\overline{x}_i$ and $f_i(x, \mathsf{u}_i(x), w) \geq 0$ when x_i reaches $\underline{x}_i$. Since system (1.1) is cooperative, we can use Proposition 1.6 with respect to the disturbance w to obtain the conditions given in Proposition 2.6. □

When $[\underline{x}, \overline{x}]$ is a robust controlled invariant, it is easy to show from Theorem 2.5 and Proposition 1.6 that for all $x \in [\underline{x}, \overline{x}]$, if $x_i = \overline{x_i}$ we have $\underline{u_i} \in \underline{U}_i(x)$ and if $x_i = \underline{x_i}$ we have $\overline{u_i} \in \overline{U}_i(x)$. Then, the necessary and sufficient conditions given by (2.3) admit a very simple realization:

$$\mathsf{u}_i(x) = \underline{u_i} + (\overline{u_i} - \underline{u_i}) \frac{\overline{x_i} - x_i}{\overline{x_i} - \underline{x_i}}. \tag{2.4}$$

The invariance controller u defined by (2.4) for all $i \in \{1, \ldots, n\}$ is affine and decentralized in the sense that the value of input $\mathsf{u}_i(x)$ only depends on state component x_i. Then, as discussed in Remark 1.13, this implies that the corresponding closed-loop system is also cooperative:

$$x \succeq_x x',\ x_i = x'_i,\ w \succeq_w w' \Rightarrow\ f_{\mathsf{u}i}(x, w) \geq f_{\mathsf{u}i}(x', w').$$

2.4 Robust local stabilizability

The notion of robust local stabilizability describes states where we can stabilize the system for any value of the disturbance. As shown later in this section, we initially considered this notion due to the fact that the characterization of a robustly locally stabilizable state is closely related to the one of robust controlled invariance from Theorem 2.5 when the interval is reduced to a single point ($\underline{x} = \overline{x}$). Let us first define this notion.

Definition 2.7 (Robust local stabilizability)**.** *The state x^* is robustly locally stabilizable if for all $\varepsilon > 0$, there exist $\delta > 0$ and $\mathsf{u} : \mathcal{B}(x^*, \varepsilon) \to [\underline{u}, \overline{u}]$ such that:*

$$\forall x_0 \in \mathcal{B}(x^*, \delta),\ \forall \mathbf{w} \in [\underline{w}, \overline{w}],\ \forall t \geq 0,\ \Phi_{\mathsf{u}}(t, x_0, \mathbf{w}) \in \mathcal{B}(x^*, \varepsilon),$$

where $\mathcal{B}(x^*, r)$ denotes the ball of radius r centered at x^*.

Definition 2.7 can be explained as follows: the target state x^* is robustly locally stabilizable if for any small ball around the state x^* there exists another ball of initial states such that the system can be robustly controlled to stay in the first ball. Thus, with a minor modification, the robust local stabilizability of x^* can be obtained with small robust controlled invariant intervals around x^*. This consideration leads to the following result.

Theorem 2.8. *Under Assumptions 1 and 2, x^* is robustly locally stabilizable if*

$$\begin{cases} f(x^*, \underline{u}, \overline{w}) \ll_x 0, \\ f(x^*, \overline{u}, \underline{w}) \gg_x 0. \end{cases}$$

If x^* is robustly locally stabilizable, then

$$\begin{cases} f(x^*, \underline{u}, \overline{w}) \preceq_x 0, \\ f(x^*, \overline{u}, \underline{w}) \succeq_x 0. \end{cases}$$

Proof. For the first implication, we choose a ball $\mathcal{B}(x^*, \varepsilon)$ of radius ε centered on x^*. Using the continuity of f with respect to the state, there exist two states $\underline{x},\ \overline{x} \in \mathcal{B}(x^*, \varepsilon)$ with $\underline{x} \ll_x x^*$ and $\overline{x} \gg_x x^*$ such that $f(\overline{x}, \underline{u}, \overline{w}) \ll_x 0$ and $f(\underline{x}, \overline{u}, \underline{w}) \gg_x 0$. Thus $[\underline{x}, \overline{x}] \subseteq \mathcal{B}(x^*, \varepsilon)$ is a robust controlled invariant interval as in Definition 2.4 and we then obtain Definition 2.7 by choosing δ such that the ball of initial states $\mathcal{B}(x^*, \delta) \subseteq [\underline{x}, \overline{x}]$.

We prove the second part of the theorem by contrapositive. Assume that there exists $i \in \{1, \dots, n\}$ such that $f_i(x^*, \underline{u}, \overline{w}) > 0$. Using the continuity of f with respect to the state, we can choose $\varepsilon > 0$ such that for all $x \in \mathcal{B}(x^*, \varepsilon),\ f_i(x, \underline{u}, \overline{w}) > 0$. If we take $w = \overline{w}$, then we can use Proposition 1.6 to extend this inequality to any u as follows:

$$\forall u \in [\underline{u}, \overline{u}],\ \exists w \in [\underline{w}, \overline{w}],\ \forall x \in \mathcal{B}(x^*, \varepsilon),\ f_i(x, u, w) > 0.$$

This means that if the state is in $\mathcal{B}(x^*, \varepsilon)$ and $w = \overline{w}$, then for any value of the control input the trajectory of the system will leave $\mathcal{B}(x^*, \varepsilon)$. This implies that x^* is not robustly locally stabilizable. This result is similarly obtained if we initially assume that there exists $i \in \{1, \dots, n\}$ such that $f_i(x^*, \overline{u}, \underline{w}) < 0$. □

Example 2.4. If we consider the temperature diffusion system (1.10) in the same conditions as in Example 2.2, we can see in Figure 2.2 that the blue and red sets are disjoints. With Theorem 2.8, this implies that in that case there exists no robustly locally stabilizable state. We can actually prove that for any choice of the control and disturbance intervals, system (1.10) can never have such states since it would require $\overline{w} < \underline{w}$.

On the other hand, the coupled-tank system (1.12) from Section 1.3.2 has some robustly locally stabilizable state. If we look at the state space of this system in Figure 2.4, the blue and red sets represent the allowed values of the lower and upper bound of a robust controlled invariant interval as in Theorem 2.5. The red set corresponds to the states where the water level can decrease with the maximal disturbance $\overline{w} = 0$ (no leak) and the minimal control $\underline{u} = (0; 0)$ (no inflow):

$$x_1 \geq 0, \qquad x_2 \geq x_1.$$

This is always true in the first dimension of (1.12), while for the second dimension we need a higher water level in tank 2 to have more outgoing flow than the incoming flow from tank 1. The blue set is obtained symmetrically by looking for the state where the water level can increase with the minimal disturbance $\underline{w} = -20$ (maximal leak) and the maximal control $\overline{u} = (22; 22)$:

$$x_1 \leq \left(\frac{K_1 \overline{u}_1}{a\sqrt{2g}}\right)^2 = 23, \qquad x_2 \leq \left(\sqrt{x_1} + \frac{K_2 \overline{u}_2 + \underline{w}}{a\sqrt{2g}}\right)^2.$$

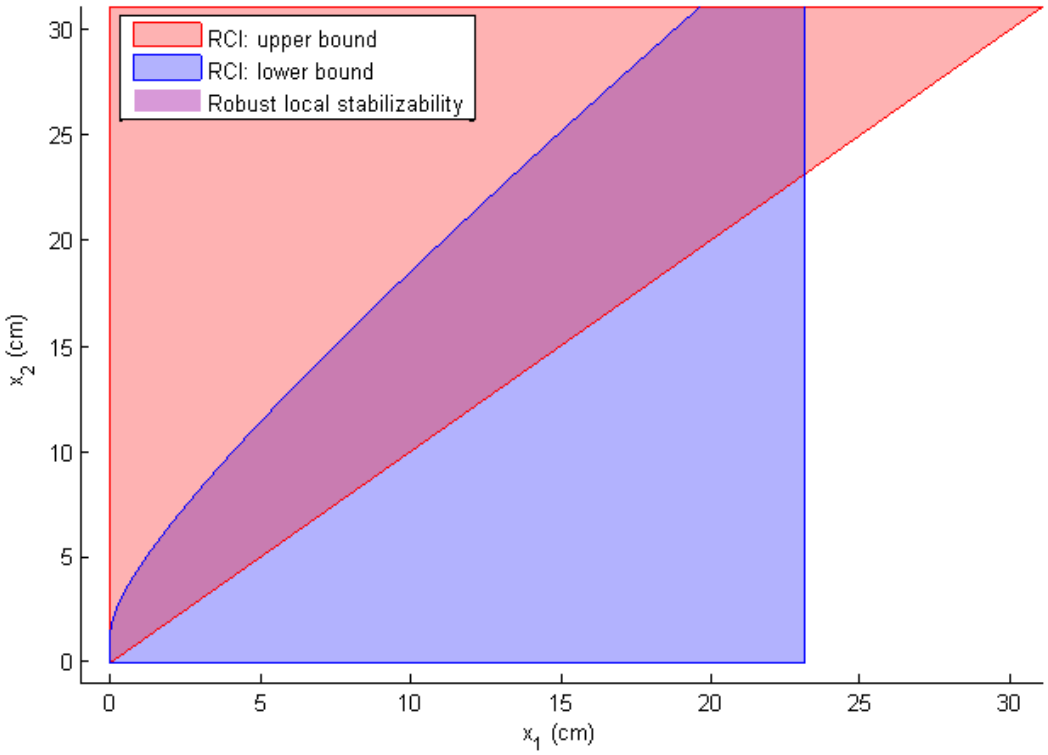


Figure 2.4 – Coupled tanks: allowed lower and upper bound for a robust controlled invariant interval (blue and red areas, respectively) and robustly locally stabilizable states (purple intersection).

For the first state, this is true when the pump inflow can compensate the outflow and for the second state the water level needs to be sufficiently small so that the outflow (from the orifice and disturbance) is smaller than the inflow (from the pump and tank 1).

The intersection of both sets (excluding their boundaries) gives the set of robustly locally stabilizable states as in Theorem 2.8, which means that we can maintain the state of the system in any small subset of the purple set from Figure 1.5. △

As mentioned in the introduction of this section, the second part of Theorem 2.8 states that having a robustly locally stabilizable state x^* implies the robust controlled invariance conditions from Theorem 2.5 for an interval reduced to a single point: $\underline{x} = \overline{x} = x^*$. However, we do not have a strict equivalence and the first implication of Theorem 2.8 requires strict inequalities as shown in the following example.

Example 2.5. Consider the system $\dot{x} = f(x, u) = x + u$ with a single state, a control input $u \in [0, 1]$ and no disturbance. For the state $x^* = 0$, the system satisfies both conditions from Theorem 2.8 with non-strict inequalities:

$$\begin{cases} f(x^*, \underline{u}) = 0 \leq 0, \\ f(x^*, \overline{u}) = 1 \geq 0. \end{cases}$$

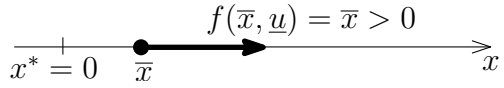


Figure 2.5 – Illustration of Example 2.5: Theorem 2.5 with non-strict inequalities.

However, as we can see in Figure 2.5, for any state $\overline{x} > x^*$ and any input $u \in [0, 1]$, the trajectory of the system initialized in $\overline{x}$ goes to infinity. Thus, according to Definition 2.7, x^* is not robustly locally stabilizable since there exists no neighborhood of x^* such that the state can be kept close to x^*. △

2.5 Robust set stabilization

In Section 2.3, we have addressed the problem of synthesizing a controller to maintain the state of system (1.1) in a given interval. The next step naturally is to look for a controller that can bring the state in this interval when the initial state lies outside the interval. For this, we use an idea similar to the robust local stabilizability in Section 2.4: we consider a family of robust controlled invariant intervals which is decreasing with respect to the set inclusion. Unlike the robust local stabilizability where we have the upper and lower bounds of the intervals converging to each other, here we are interested in stabilization in a set and the family of intervals converges toward a robust controlled invariant interval.

For a general definition, let $\mathcal{S}_0$ be a set of initial states and $\mathcal{S}$ the target set where we shall steer the state of system (1.1). The robust set stabilization from $\mathcal{S}_0$ to $\mathcal{S}$ is possible in finite time if there exists a stabilizing controller as defined below.

Definition 2.9 (Stabilizing controller)**.** *A controller* $\mathsf{u} : \mathcal{S}_0 \to [\underline{u}, \overline{u}]$ *is said to be a stabilizing controller from* $\mathcal{S}_0$ *to* $\mathcal{S}$ *if*

$$\forall x_0 \in \mathcal{S}_0,\ \forall \mathbf{w} \in [\underline{w}, \overline{w}],\ \exists T \geq 0 \mid \forall t \geq T,\ \ \Phi_{\mathsf{u}}(t, x_0, \mathbf{w}) \in \mathcal{S}.$$

As said above, we are interested in working with intervals to use the results on robust controlled invariance from Theorem 2.5. Let $[\underline{x_0}, \overline{x_0}]$ be an interval of initial states and $[\underline{x_f}, \overline{x_f}] \subseteq [\underline{x_0}, \overline{x_0}]$ a target interval such that $\underline{x_0} \ll_x \underline{x_f}$ and $\overline{x_f} \ll_x \overline{x_0}$. We aim to synthesize stabilizing controllers from $[\underline{x_0}, \overline{x_0}]$ to $[\underline{x_f}, \overline{x_f}]$ under the following assumption.

Assumption 4. *There exist continuously differentiable functions*

$$\underline{X},\ \overline{X} : [0, 1] \to \mathbb{R}^n,$$

respectively strictly decreasing and increasing on all their components

$$\frac{d\underline{X}}{d\lambda}(\lambda) \ll_x 0,\ \ \frac{d\overline{X}}{d\lambda}(\lambda) \gg_x 0,\ \ \forall \lambda \in [0, 1], \tag{2.5}$$

such that $\underline{X}(0) = \underline{x_f}$, $\underline{X}(1) = \underline{x_0}$, $\overline{X}(0) = \overline{x_f}$, $\overline{X}(1) = \overline{x_0}$ *and satisfying*

$$f(\underline{X}(\lambda), \overline{u}, \underline{w}) \gg_x 0,\ \ f(\overline{X}(\lambda), \underline{u}, \overline{w}) \ll_x 0,\ \ \forall \lambda \in [0, 1]. \tag{2.6}$$

The functions $\underline{X}$ and $\overline{X}$ will serve as support for the lower and upper bounds of the robust controlled invariant intervals used for the stabilization from $[\underline{x_0}, \overline{x_0}] = [\underline{X}(1), \overline{X}(1)]$ to $[\underline{x_f}, \overline{x_f}] = [\underline{X}(0), \overline{X}(0)]$. In Section 2.5.1, we present a method to synthesize a stabilizing controller under Assumption 4, while in Section 2.5.2 we give several examples of support functions $\underline{X}$ and $\overline{X}$ satisfying Assumption 4.

2.5.1 Stabilizing controller synthesis

The last condition (2.6) of Assumption 4 and Theorem 2.5 imply that the interval $[\underline{X}(\lambda), \overline{X}(\lambda')]$ is a robust controlled invariant for all $\lambda, \lambda' \in [0, 1]$. In addition, since (2.6) involves strict inequalities we know that when the state is on the boundary of an interval $[\underline{X}(\lambda), \overline{X}(\lambda')]$, not only we can keep the state in this interval, but we can also force it toward the interior. The main idea of our approach is thus to use this parameterized family of robust controlled invariants to drive the state to $[\underline{x_f}, \overline{x_f}]$. Let us reformulate some of the conditions in Assumption 4 in a way that will be useful for the proof to come and to give an upper bound on the stabilization time.

Remark 2.10. *From Assumption 4, there exists $\alpha > 0$ such that for all $i \in \{1, \dots, n\}$, $\lambda \in [0, 1]$,*

$$f_i(\underline{X}(\lambda), \overline{u}, \underline{w}) \geq \alpha \text{ and } f_i(\overline{X}(\lambda), \underline{u}, \overline{w}) \leq -\alpha. \tag{2.7}$$

Since $\overline{X}$ is strictly increasing with $\frac{d\overline{X}}{d\lambda}(\lambda) \gg_x 0$ and continuously differentiable, then $\overline{X}_i^{-1}$ is well defined, strictly increasing and continuously differentiable on $[\overline{x_f}_i, \overline{x_0}_i]$. Similarly, $\underline{X}_i^{-1}$ is well defined, strictly decreasing and continuously differentiable on $[\underline{x_0}_i, \underline{x_f}_i]$. It follows that there exists $\beta > 0$ such that for all $i \in \{1, \dots, n\}$,

$$\begin{cases} \forall x_i \in [\overline{x_f}_i, \overline{x_0}_i], \ \dfrac{d}{dx_i}\overline{X}_i^{-1}(x_i) \geq \beta \\ \forall x_i \in [\underline{x_0}_i, \underline{x_f}_i], \ \dfrac{d}{dx_i}\underline{X}_i^{-1}(x_i) \leq -\beta. \end{cases} \tag{2.8}$$

Under Assumption 4, we define the functions $\underline{\lambda}, \ \overline{\lambda} : [\underline{x_0}, \overline{x_0}] \to [0, 1]$ as

$$\begin{cases} \overline{\lambda}(x) = \min\{\lambda \in [0, 1] \mid \overline{X}(\lambda) \succeq_x x\}, \\ \underline{\lambda}(x) = \min\{\lambda \in [0, 1] \mid \underline{X}(\lambda) \preceq_x x\}. \end{cases} \tag{2.9}$$

In other words, $[\underline{X}(\underline{\lambda}(x)), \overline{X}(\overline{\lambda}(x))]$ is the smallest interval of the parameterized family $[\underline{X}(\lambda), \overline{X}(\lambda')]$ containing x. This interval is illustrated in Figure 2.6 for two possible positions of the state x. An alternative expression of $\underline{\lambda}$ and $\overline{\lambda}$ can be obtained by assuming that the domain of definition of the functions $\overline{X}_i^{-1}$ and $\underline{X}_i^{-1}$ can be extended to $[\underline{x_0}_i, \overline{x_0}_i]$ while keeping their properties of continuous differentiability and strict monotonicity. This means that $\overline{X}_i^{-1}$ and $\underline{X}_i^{-1}$ take negative values for $x_i < \overline{x_f}_i$ and $x_i > \underline{x_f}_i$, respectively. If we introduce the functions $\underline{\lambda}_i, \ \overline{\lambda}_i : [\underline{x_0}, \overline{x_0}] \to [0, 1]$ such that for all $x \in [\underline{x_0}, \overline{x_0}]$ and $i \in \{1, \dots, n\}$,

$$\begin{cases} \overline{\lambda}_0(x) = 0 \text{ and } \overline{\lambda}_i(x) = \overline{X}_i^{-1}(x_i), \\ \underline{\lambda}_0(x) = 0 \text{ and } \underline{\lambda}_i(x) = \underline{X}_i^{-1}(x_i), \end{cases}$$

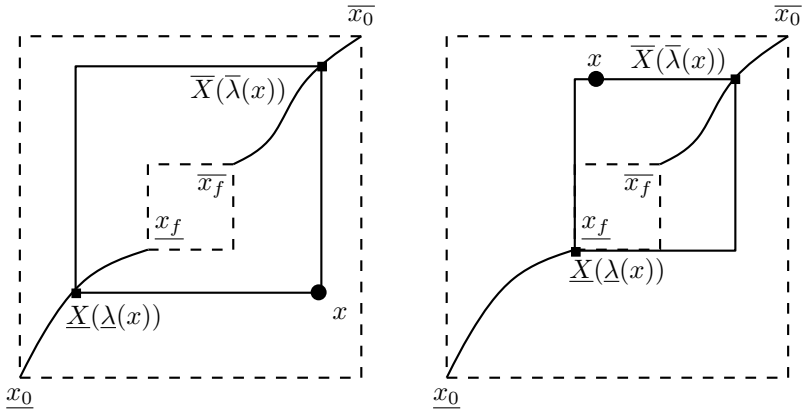


Figure 2.6 – Smallest element $[\underline{X}(\underline{\lambda}(x)), \overline{X}(\overline{\lambda}(x))]$ of the parameterized family of robust controlled invariant intervals $[\underline{X}(\lambda), \overline{X}(\lambda')]$ containing state x.

then the functions $\underline{\lambda}$ and $\overline{\lambda}$ in (2.9) can now be written as the maximum of continuously differentiable functions:

$$\begin{cases} \overline{\lambda}(x) = \max\limits_{i\in\{0,\dots,n\}} \overline{\lambda}_i(x), \\ \underline{\lambda}(x) = \max\limits_{i\in\{0,\dots,n\}} \underline{\lambda}_i(x). \end{cases}$$

Note that this extension of the domain of definition of $\overline{X}_i^{-1}$ and $\underline{X}_i^{-1}$ is not included in Assumption 4 since it is not necessary for the robust set stabilization. It is introduced because it simplifies the notations of $\underline{\lambda}$ and $\overline{\lambda}$ and the proof of Theorem 2.11.

The main idea of our stabilization approach is to use a feedback control u that renders each interval $[\underline{X}(\underline{\lambda}(x)), \overline{X}(\overline{\lambda}(x))]$ robust invariant for the closed-loop system (1.2). This control strategy makes $\underline{\lambda}(x)$ and $\overline{\lambda}(x)$ act like Lyapunov functions which are then used to show that the state reaches the target interval $[\underline{x_f}, \overline{x_f}] = [\underline{X}(0), \overline{X}(0)]$ in finite time. Note that we can obtain a controller rendering the intervals $[\underline{X}(\underline{\lambda}(x)), \overline{X}(\overline{\lambda}(x))]$ invariant by adapting the simple affine controller (2.4) as follows:

$$\mathsf{u}_i(x) = \underline{u_i} + (\overline{u_i} - \underline{u_i}) \frac{\overline{X}_i(\overline{\lambda}(x)) - x_i}{\overline{X}_i(\overline{\lambda}(x)) - \underline{X}_i(\underline{\lambda}(x))}. \tag{2.10}$$

Theorem 2.11. *Under Assumptions 1, 2 and 4, the controller* u *defined by (2.9) and (2.10) is a stabilizing controller from* $[\underline{x_0}, \overline{x_0}]$ *to* $[\underline{x_f}, \overline{x_f}]$.

Proof. Let $\overline{\mathcal{I}}(x) = \{i \in \{0,\dots,n\} \mid \overline{\lambda}_i(x) = \overline{\lambda}(x)\}$. Let $x_0 \in [\underline{x_0}, \overline{x_0}]$, $\mathbf{w} \in [\underline{w}, \overline{w}]$, $\mathbf{x} = \Phi_{\mathsf{u}}(., x_0, \mathbf{w})$, $t \in \mathbb{R}_0^+$ and $i \in \overline{\mathcal{I}}(\mathbf{x}(t)) \backslash \{0\}$. We defined $\overline{\lambda}_i(x) = \overline{X}_i^{-1}(x_i)$, which implies:

$$\frac{d\overline{\lambda}_i}{dt}(\mathbf{x}(t)) = \frac{d\overline{X}_i^{-1}}{dx_i}(\mathbf{x}_i(t)) * f_i(\mathbf{x}(t), \mathsf{u}_i(\mathbf{x}(t)), \mathbf{w}(t)).$$

Since $i \in \overline{\mathcal{I}}(\mathbf{x}(t))$, we have $\mathbf{x}_i(t) = \overline{X}_i(\overline{\lambda}(\mathbf{x}(t)))$ and (2.10) gives $\mathsf{u}_i(\mathbf{x}(t)) = \underline{u_i}$. Then we can obtain:

$$f_i(\mathbf{x}(t), \mathsf{u}_i(\mathbf{x}(t)), \mathbf{w}(t)) \leq f_i(\overline{X}(\overline{\lambda}(\mathbf{x}(t))), \underline{u_i}, \overline{w}) \leq -\alpha.$$

by using Proposition 1.6 for the first inequality and (2.7) for the second one. Inequalities (2.8) then imply that $\frac{d}{dt}\overline{\lambda}_i(\mathbf{x}(t)) \leq -\alpha\beta$ for all i in $\overline{\mathcal{I}}(\mathbf{x}(t))\backslash\{0\}$. Since $\overline{\lambda}(x) = \max_{i\in\{0,\dots,n\}}(\overline{\lambda}_i(x))$, where the functions $\overline{\lambda}_i$ are continuously differentiable, its upper right Dini derivative is given by [BM07]:

$$D^+\overline{\lambda}(\mathbf{x}(t)) = \max_{i\in\overline{\mathcal{I}}(\mathbf{x}(t))} \frac{d\overline{\lambda}_i}{dt}(\mathbf{x}(t)).$$

When $\overline{\lambda}(\mathbf{x}(t)) > 0$, the index 0 is not in $\overline{\mathcal{I}}(\mathbf{x}(t))$ and $\overline{\lambda}$ is strictly decreasing ($D^+\overline{\lambda}(\mathbf{x}(t)) \leq -\alpha\beta$) and thus acts like a Lyapunov function. When $\overline{\lambda}(\mathbf{x}(t)) = 0$, we have $0 \in \overline{\mathcal{I}}(\mathbf{x}(t))$ and $D^+\overline{\lambda}(\mathbf{x}(t)) = 0$, hence if the state is in the target interval, it remains in it. From what precedes and [BM07], we can integrate the Dini derivative between the initial time and the first instant $\bar{t}$ such that $\overline{\lambda}(\mathbf{x}(\bar{t})) = 0$,

$$\overline{\lambda}(\mathbf{x}(\bar{t})) - \overline{\lambda}(\mathbf{x}(0)) = \int_0^{\bar{t}} D^+\overline{\lambda}(\mathbf{x}(s))ds \leq -\alpha\beta\bar{t},$$

which then implies:

$$\forall t \geq \frac{\overline{\lambda}(\mathbf{x}(0))}{\alpha\beta},\ \overline{\lambda}(\mathbf{x}(t)) = 0.$$

Similarly, we can show that $\underline{\lambda}(\mathbf{x}(t)) = 0$ for all $t \geq \underline{\lambda}(\mathbf{x}(0))/\alpha\beta$. Thus u is a stabilizing controller. □

The proof of Theorem 2.11 is presented with a particular stabilizing controller given by (2.10) but there exist many other stabilizing controllers. It is for instance sufficient to choose the control input $\mathsf{u}(x)$ such that the functions $\underline{\lambda}(x)$ and $\overline{\lambda}(x)$ defined by (2.9) are strictly decreasing. Also, even though (2.10) is based on the affine and decentralized controller (2.4), this stabilizing controller is neither affine nor decentralized. Note that the maximal stabilization time $1/\alpha\beta$ may be tuned by a suitable choice of $\underline{X}$ and $\overline{X}$ (see (2.7) and (2.8)).

Remark 2.12. *Let a state x^* satisfy the conditions for robust local stabilizability from Theorem 2.8. Then for any small neighborhood $[\underline{x_f}, \overline{x_f}]$ of x^* satisfying Assumption 4, the controller u defined by (2.9) and (2.10) is a stabilizing controller from $[\underline{x_0}, \overline{x_0}]$ to $[\underline{x_f}, \overline{x_f}]$.*

2.5.2 Choice of the support functions

The result presented in the previous section is based on the existence of two support functions $\underline{X}$ and $\overline{X}$ such that Assumption 4 holds. In the following, we describe three possible choices of such functions and some conditions to ensure the satisfaction of Assumption 4. Let us first remind that Assumption 4 can be split into its three main conditions:

- $\underline{X}$, $\overline{X} : [0,1] \to \mathbb{R}^n$ are continuously differentiable with $\underline{X}(0) = \underline{x_f}$, $\underline{X}(1) = \underline{x_0}$, $\overline{X}(0) = \overline{x_f}$, $\overline{X}(1) = \overline{x_0}$;
- (2.5): they are respectively strictly decreasing and increasing on all their components;

- (2.6): all intervals $[\underline{X}(\lambda), \overline{X}(\lambda')]$ satisfy the robust controlled invariance conditions from Theorem 2.5 with strict inequalities.

Based on the stabilization method in the previous section and condition (2.6) for $\lambda = 1$ and $\lambda = 0$, we are only interested in robust set stabilization between two robust controlled invariant intervals $[\underline{x_0}, \overline{x_0}]$ and $[\underline{x_f}, \overline{x_f}]$ with $\underline{x_0} \ll_x \underline{x_f}$ and $\overline{x_f} \ll_x \overline{x_0}$. Thus we assume that we have such intervals. We also consider that Assumptions 1 and 2 are satisfied since they are required in Theorem 2.11.

Linear functions The first possible choice is to consider the simple linear functions:

$$\begin{cases} \overline{X}(\lambda) = \lambda \overline{x_0} + (1-\lambda)\overline{x_f}, \\ \underline{X}(\lambda) = \lambda \underline{x_0} + (1-\lambda)\underline{x_f}. \end{cases} \tag{2.11}$$

The first condition of Assumption 4 is immediately satisfied. The second condition (2.5) is a direct implication of the assumption that $\underline{x_0} \ll_x \underline{x_f}$ and $\overline{x_f} \ll_x \overline{x_0}$. With the functions (2.11), the last condition (2.6) is not always satisfied and depends on the dynamics of the system. If the system (1.1) is such that the sets

$$\{x \in \mathbb{R}^n |\ f(x, \overline{u}, \underline{w}) \gg_x 0\} \text{ and } \{x \in \mathbb{R}^n |\ f(x, \underline{u}, \overline{w}) \ll_x 0\}$$

are convex, then (2.6) is automatically satisfied since $\overline{X}(\lambda)$ is a convex combination of $\overline{x_0}$ and $\overline{x_f}$ (and similarly for $\underline{X}(\lambda)$ with $\underline{x_0}$ and $\underline{x_f}$). Otherwise (2.6) needs to be checked and the simple form of the functions (2.11) allows an easy numerical verification of this condition.

Remark 2.13. *Under Assumption 3, we know from Proposition 1.6 and the definition of the static input-state map k_x that the minimal robust invariant interval $[k_x(\underline{u}, \underline{w}), k_x(\overline{u}, \overline{w})]$ from Theorem 2.3 is a robust controlled invariant. If in addition this interval satisfies the robust controlled invariance with strict inequalities, we can start the stabilization in $[\underline{x_0}, \overline{x_0}] = [k_x(\underline{u}, \underline{w}), k_x(\overline{u}, \overline{w})]$.*

Example 2.6. We consider the same conditions as in Example 2.2 with an initial state $x_0 = (26; 17)$ chosen outside the robust controlled invariant interval. We apply the robust set stabilization method with the support functions from (2.11) and a set of initial states $[\underline{x_0}, \overline{x_0}]$ equal to the minimal robust invariant interval from Theorem 2.3. As it can be seen on Figure 2.7, the lower and upper bounds of this interval (dashed on the figure) are on the boundary of the blue and red subsets, respectively. This means that the chosen interval of initial states $[\underline{x_0}, \overline{x_0}]$ does not satisfy the robust controlled invariance with strict inequalities. Since we can see that all the remaining points of the support functions satisfy this condition, we can keep this interval and simply make sure that we only take initial states in its interior.

The controller (2.10) is obtained as follows. First we compute the projections of x on the support function along each dimension:

$$\overline{\lambda}_1(x) = \overline{X_1}^{-1}(x_1) = \frac{x_1 - \overline{x_f}_1}{\overline{x_0}_1 - \overline{x_f}_1} \quad \text{and} \quad \overline{\lambda}_2(x) = \overline{X_2}^{-1}(x_2) = \frac{x_2 - \overline{x_f}_2}{\overline{x_0}_2 - \overline{x_f}_2}.$$

Then $\overline{\lambda}(x) = max(0, \overline{\lambda}_1(x), \overline{\lambda}_2(x))$ and $\overline{X_1}(\overline{\lambda}(x)) = \overline{\lambda}(x)\overline{x_0}_1 + (1-\overline{\lambda}(x))\overline{x_f}_1$. Similarly, we compute $\underline{X_1}(\underline{\lambda}(x))$ and we can apply the control (2.10) where $\mathsf{u}_1 = \mathsf{u}$ and $\mathsf{u}_2 = \emptyset$.

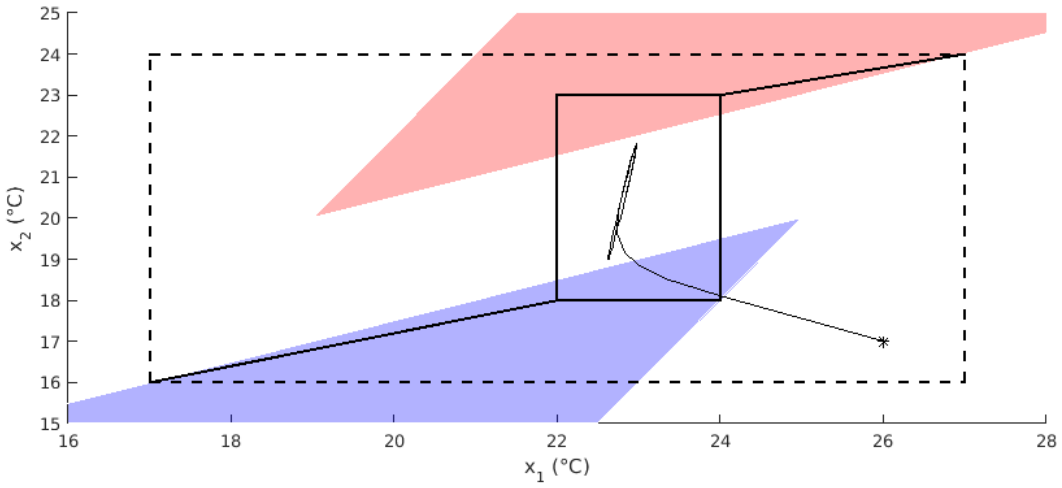


Figure 2.7 – Example 2.6: robust set stabilization with linear support functions.

We can see in the example of Figure 2.7 that this controller correctly stabilizes the state into the black robust controlled invariant interval. We can note that as soon as the state x enters the target interval, we have $\overline{\lambda}(x) = \underline{\lambda}(x) = 0$ and the stabilization controller (2.10) becomes the simpler decentralized and affine controller (2.4), which is already known to maintain the state in a robust controlled invariant interval. △

Families of equilibria The second possible choice that we present is based on the static input-state map from Assumption 3. We consider that the bounds of $[\underline{x_0}, \overline{x_0}]$ and $[\underline{x_f}, \overline{x_f}]$ can be described as the following equilibria.

Assumption 5. *Under Assumption 3, there exists* $\underline{u_0}$, $\underline{u_f}$, $\overline{u_0}$, $\overline{u_f} \in [\underline{u}, \overline{u}]$ *such that:*

$$\begin{cases} \underline{u} \ll_u \overline{u_f} \ll_u \overline{u_0}, & \overline{x_0} = k_x(\overline{u_0}, \overline{w}), \quad \overline{x_f} = k_x(\overline{u_f}, \overline{w}), \\ \underline{u_0} \ll_u \underline{u_f} \ll_u \overline{u}, & \underline{x_0} = k_x(\underline{u_0}, \underline{w}), \quad \underline{x_f} = k_x(\underline{u_f}, \underline{w}). \end{cases}$$

We can now define the support functions $\underline{X}$ and $\overline{X}$ as equilibria using a convex combination of the control inputs $\underline{u_0}$, $\underline{u_f}$, $\overline{u_0}$, $\overline{u_f}$ from Assumption 5:

$$\begin{cases} \overline{U}(\lambda) = \lambda \overline{u_0} + (1-\lambda)\overline{u_f}, & \overline{X}(\lambda) = k_x(\overline{U}(\lambda), \overline{w}), \\ \underline{U}(\lambda) = \lambda \underline{u_0} + (1-\lambda)\underline{u_f}, & \underline{X}(\lambda) = k_x(\underline{U}(\lambda), \underline{w}). \end{cases} \tag{2.12}$$

Proposition 2.14. *Let Assumptions 1, 2, 3, 5 hold and further assume that*

- *the vector field f of system (1.1) is continuously differentiable;*
- *its matrix of partial derivatives $\partial f / \partial x$ is invertible;*
- *$\partial f_i / \partial u_i > 0$ for all $i \in \{1, \dots, n\}$, where u_i denotes the vector of control inputs directly influencing the state x_i as in Definition 1.12 for the local control.*

Then the functions $\underline{X}$ and $\overline{X}$ defined by (2.12) satisfy Assumption 4.

Proof. Here again, the first condition of Assumption 4 is satisfied by the definition of the support functions. From Remark 1.15, we know that k_x is monotone. It is thus straightforward to show that $\underline{X}$ and $\overline{X}$ are decreasing and increasing, respectively. Moreover, note that $f(\underline{X}(\lambda), \underline{U}(\lambda), \underline{w}) = 0$. By the implicit functions theorem it follows that $\underline{X}$ is continuously differentiable and that

$$\frac{\partial f}{\partial x}(\underline{X}(\lambda), \underline{U}(\lambda), \underline{w}) \times \frac{d\underline{X}}{d\lambda}(\lambda) = -\frac{\partial f}{\partial u}(\underline{X}(\lambda), \underline{U}(\lambda), \underline{w}) \times \frac{d\underline{U}}{d\lambda}(\lambda).$$

With Assumptions 1, 2 and $\underline{u_0} \ll_u \underline{u_f}$ from Assumption 5, we have that for all $i \in \{1, \dots, n\}$:

$$\sum_{j=1}^{n} \frac{\partial f_i}{\partial x_j}(\underline{X}(\lambda), \underline{U}(\lambda), \underline{w}) \times \frac{d\underline{X}_j}{d\lambda}(\lambda) = -\frac{\partial f_i}{\partial u_i}(\underline{X}(\lambda), \underline{U}(\lambda), \underline{w}) \times \frac{d\underline{U}_i}{d\lambda}(\lambda) > 0.$$

Then with $\underline{X}$ decreasing and Assumption 1, it yields

$$\frac{\partial f_i}{\partial x_i}(\underline{X}(\lambda), \underline{U}(\lambda), \underline{w}) \times \frac{d\underline{X}_i}{d\lambda}(\lambda) > -\sum_{j \neq i} \frac{\partial f_i}{\partial x_j}(\underline{X}(\lambda), \underline{U}(\lambda), \underline{w}) \times \frac{d\underline{X}_j}{d\lambda}(\lambda) \geq 0,$$

which implies that $\frac{d\underline{X}_i}{d\lambda}(\lambda) \neq 0$, hence $\underline{X}$ is strictly decreasing on all its components. Similarly, we can show that $\overline{X}$ is continuously differentiable and strictly increasing on all its components: (2.5) thus holds. Lastly, (2.6) is obtained by considering $\partial f_i / \partial u_i > 0$ from the proposition statement and $\underline{U}(\lambda) \ll_u \overline{u}$ and $\overline{U}(\lambda) \gg_u \underline{u}$ from (2.12) and Assumption 5, which give for all $\lambda \in [0, 1]$:

$$\begin{cases} f(\overline{X}(\lambda), \underline{u}, \overline{w}) \ll_x f(\overline{X}(\lambda), \overline{U}(\lambda), \overline{w}) = 0, \\ f(\underline{X}(\lambda), \overline{u}, \underline{w}) \gg_x f(\underline{X}(\lambda), \underline{U}(\lambda), \underline{w}) = 0. \end{cases}$$

□

Trajectories between equilibria The last example of possible support functions is close to the previous one in terms of assumptions and has the advantage of being easier to create numerically. As for the functions (2.12), we consider that Assumptions 3 and 5 hold: the interval bounds $\underline{x_0}$, $\underline{x_f}$, $\overline{x_f}$ and $\overline{x_0}$ can be characterized as equilibria. The idea is to consider the trajectories between $\underline{x_0}$ and $\underline{x_f}$ and between $\overline{x_f}$ and $\overline{x_0}$ with the corresponding pair of constant inputs (u, w) from Assumption 5. We can then parameterize the support function by the following trajectories of the system:

$$\begin{cases} \overline{X}(\lambda) = \Phi(\frac{\lambda}{1-\lambda}, \overline{x_f}, \overline{u_0}, \overline{w}), \\ \underline{X}(\lambda) = \Phi(\frac{\lambda}{1-\lambda}, \underline{x_f}, \underline{u_0}, \underline{w}). \end{cases} \tag{2.13}$$

Similarly, we can define trajectories in the opposite direction:

$$\begin{cases} \overline{X}(\lambda) = \Phi(\frac{1-\lambda}{\lambda}, \overline{x_0}, \overline{u_f}, \overline{w}), \\ \underline{X}(\lambda) = \Phi(\frac{1-\lambda}{\lambda}, \underline{x_0}, \underline{u_f}, \underline{w}). \end{cases} \tag{2.14}$$

For non-linear systems, numerically solving the static input-state map equations to obtain the equilibria corresponding to a few conditions can usually be done relatively easily. On the other hand, characterizing the whole family of equilibria as in (2.12) may be much harder than simply computing trajectories as in (2.13) and (2.14). However, the simplified numerical implementation of these support functions comes with a tradeoff that (2.13) and (2.14) do not necessarily satisfy Assumption 4. Conditions (2.5) and (2.6) from Assumption 4 thus need to be verified numerically. Note that the parameterization of the time by $t(\lambda) = \frac{\lambda}{1-\lambda}$ or $t(\lambda) = \frac{1-\lambda}{\lambda}$ can be replaced by any strictly monotone function $t : [0,1] \to \mathbb{R}_0^+ \cup \{+\infty\}$ with $t(0) = 0$ and $t(\lambda) \underset{\lambda \to 1}{\longrightarrow} +\infty$ for (2.13) or with $t(1) = 0$ and $t(\lambda) \underset{\lambda \to 0}{\longrightarrow} +\infty$ for (2.14).

Chapter 3

Symbolic control of cooperative systems

In this chapter, we are interested in synthesizing a controller for a continuous system (1.1) using symbolic methods. The starting point of this approach is to create a finite abstraction of the continuous behavior. The obtained discrete system is called a symbolic abstraction as its states can be seen as symbols representing infinitely many states of the continuous system. The purely discrete nature of the symbolic abstraction allows the use of well established controller synthesis techniques to realize complex specifications. If some behavioral relationship relates the dynamics of the continuous and symbolic models, the discrete controller synthesized for the symbolic abstraction can be refined into a controller for the original system.

This chapter is organized as follows. We give an overview of the main literature on symbolic abstraction and symbolic control and motivate the choice of this approach in Section 3.1. The notations and definitions that are used in this chapter, mainly based on those introduced in Paulo Tabuada's book [Tab09], are presented in Section 3.2, followed by the formulation of our control problem. In Section 3.3, we create a symbolic abstraction of the cooperative system (1.1). Then, this abstraction is used in Section 3.4 to synthesize a controller realizing a safety specification in an interval (1.3). Since the obtained controller may include several safety strategies, we choose the strategy that is optimal according to a particular performance criterion by using a receding horizon control scheme on the result of a dynamic programming algorithm. Finally, in Section 3.5 we provide performance guarantees based on the optimization run on the safe control strategies.

An experimental validation of this method for the temperature control on the system described in Chapter 5 has been given in [MGW15].

3.1 Motivations and related work

Model simplification When dealing with the problem of controller synthesis for complex dynamical systems, possibly exhibiting non-linear or hybrid behaviors, the classical and well established results on control of continuous linear systems [TSH01] are not applicable. Even though other more robust or adaptive methods such as

$\mathcal{H}_\infty$ control [SP05] or model predictive control [RM09] exist in the field of control of continuous systems, it may be interesting instead to look at controller synthesis on a simplified version of our system. A simplified model describing the same system can be created for various reasons such as reducing the dimension of the system [GP09], lowering the level of precision of the model by neglecting details that are not useful or not available [Lun94] and abstracting undesirable non-linear or hybrid dynamics into a purely discrete model [AHLP00]. These methods have in common that they allow the creation of an abstraction of the original system for which the controller synthesis problem is easier.

Behavioral relationship In order to synthesize a controller on an abstraction and refine it into a controller for the original system, we require that some formal behavioral relationship exists between the two models. The purpose of such relationships is to ensure that the original system and its abstraction behave in a similar way. The most basic relationship is that of language or behavioral inclusion: assuming that both models are observable through a common output space, any sequence observed on the original model can be observed on its abstraction. This means that the set of outputs that are reachable from the abstraction contains the reachable outputs of the original system. We refer to language or behavioral equivalence when we also have a similar inclusion from the abstraction to the original system. This type of inclusion or equivalence is computationally expensive to check, even on finite transition systems. That is why in most cases we rather consider stronger notions that are easier to prove, such as simulation, bisimulation and their alternating and approximate versions [Tab09]. When both compared models are described as (possibly infinite) transition systems, the general idea of a simulation relation is that any transition of the original system is matched by a transition of the abstraction. Such simulation relation naturally implies the behavioral or language inclusion. A bisimulation relation is obtained when the original system also simulates the abstraction, which thus implies the behavioral equivalence. When dealing with control systems, as it is the case in this chapter, we use the notion of alternating simulation, where we investigate the existence of control actions on the original system enforcing a desired behavior: for any control taken in the abstraction there exists a control of the original system such that the transitions of the latter are matched by transitions of the former. These exact relationships may be too restrictive in some cases and are not robust to unmodeled disturbances. The notion of approximate simulation thus has been introduced to consider relationships between systems whose behaviors are not identical but remain at a distance smaller than some chosen precision [GP07].

Symbolic abstraction The methods leading to a discrete abstraction of the original system can be obtained in several ways. In [AHLP00], an initial partition of the state space is obtained using an equivalence relation, then a bisimulation algorithm is applied to refine this partition by splitting its elements based on backward reachable sets. Another method based on a finite partition (or a finite covering) of the state space is to compute or approximate the reachable sets of the elements of the partition [Rei09]. If instead of a partition we use a quantization of the state space, we consider that the behavior of an element of the abstraction approximates

those of all continuous state in its neighborhood [PGT08]. In all these cases, each element of the discrete abstraction can be seen as a symbol representing infinitely many states of the continuous system, which explains the denomination of *symbolic abstraction*.

In this chapter, we focus on methods where the symbolic abstraction is a finite transition system obtained from a partition of the state space and considerations on the reachable set of each symbol. For this abstraction to satisfy a behavioral relationship as described above, we need to create its transitions based on the dynamics of the original system. This is achieved by computing the reachable set of symbols for a given control action and sampling period and taking the intersection of this set with the partition. Since infinitely many states are aggregated into a single symbol, the obtained abstraction is a non-deterministic transition system. In most cases, the exact computation of a reachable set cannot be achieved and we rely on approximations. To ensure the inclusion of all the behaviors of the original system in those of its abstraction, we necessarily need to consider over-approximations of the reachable set, which also prevents the possibility of obtaining a behavioral equivalence. The reachable set can be obtained in several ways, using for example polytopes [CK99], oriented hyper-rectangles [SK03], ellipsoids [KV07], zonotopes [GLG08] or level sets [MT00]. For systems satisfying the monotonicity property as described in Chapter 1, hyper-rectangle over-approximations are particularly easy to obtain [MR02]. This method can also be extended to the class of mixed-monotone systems [CA15].

Symbolic control Another advantage of creating purely discrete abstractions of continuous or hybrid systems is that of allowing the use of discrete synthesis techniques such as those in the domains of supervisory control [RW87] or game theory [PPS06]. In addition, while continuous control theory usually focuses on traditional properties such as stability, observability or controllability, these discrete techniques can address more complex specifications describing the desired behavior of the controlled system over time, formulated as automata [CL08] or temporal logic formulas [Pnu77]. Temporal logic is a rich specification language combining logical operators (e.g. *not*, *and*, *or*) with temporal operators (e.g. *always*, *eventually*, *until*) which covers the needs of a wide variety of applications. To compare our results with those on robust controlled invariance presented in Chapter 2, we only focus on safety, which is one of the simplest temporal logic specification: the state of the system must always remain in the safe set. Note that knowing the specification beforehand is essential when creating a symbolic abstraction since it has a significant influence on where the focus should be and what information can be abstracted. For example, with a safety specification, a fine partition of the state space outside of the safe set is not useful since the goal is to forbid all the transitions leading there. Control strategies realizing a given specification are not necessarily unique and we can then choose among the allowed strategies the one that is optimal according to some performance criterion complementing the specification.

Robustness There are two main challenges in the field of symbolic control: scalability and robustness. The scalability problem is addressed in Chapter 4. The ro-

bustness issue appears when the original system is subject to external disturbances or modeling errors. When the abstraction is created from the nominal conditions of this system, the real disturbed system may generate some behaviors that have no equivalence in the abstraction. The behavioral inclusion thus may be lost if these disturbances are not taken into account in the abstraction. The approximate simulation [GP07] already described above is the first possibility to approach this robustness problem since it relaxes the exact behavioral inclusions to allow slight mismatches between the abstraction and the possibly disturbed original system. A second approach, used in this chapter, is to create the abstraction from a model that already includes the effect of the disturbances. Ensuring the behavioral inclusion requires to consider the worst cases of the disturbances in the abstraction, which increases the non-determinism but keeps providing controllers that are correct by construction as long as all disturbances are correctly modeled and remain in their estimated bounds. The third approach, with similar consequences, is to include an estimate of the disturbance bounds directly in the abstraction [LO14]. The last approach is inspired by the notion of robustness considered in continuous control and more precisely input-output stability [TCRM14]. In this method, the synthesized controllers are correct by construction for the nominal case without disturbance and bounded disturbances implies a bounded deviation to the desired behavior. Note that the advantage of this method is that the disturbances do not need to be modeled or estimated.

3.2 Preliminaries

In this section, we recall some notations and definitions from [Tab09] adapted to fit our particular conditions and present the control problem that we want to solve.

3.2.1 Definitions

Let us start by the general definition of a system formulated as a transition system.

Definition 3.1 (System)**.** *A system is a quadruple* $S = (X, X^0, U, \longrightarrow)$ *consisting of the following elements:*

- *a set of states* X*,*
- *a set of initial states* $X^0 \subseteq X$*,*
- *a set of inputs* U*,*
- *a transition relation* $\longrightarrow \subseteq X \times U \times X$*.*

A transition $(x, u, x') \in \longrightarrow$ *of* S *is equivalently written as* $x \xrightarrow{u} x'$ *or* $x' \in Post(x, u)$*. The set* $U(x) \subseteq U$ *denotes the set of inputs* u *such that* $Post(x, u) \neq \emptyset$*. A trajectory of* S *is an infinite sequence* $(x^0, u^0, x^1, u^1, \dots)$ *such that* $x^0 \in X^0$ *and for all* $k \in \mathbb{N}$*,* $u^k \in U(x^k)$ *and* $x^{k+1} \in Post(x^k, u^k)$*.*

Remark 3.2. *For systems with no constraint on the inputs, as it is the case in this thesis when no feedback control is applied to the system, we have* $U(x) = U$ *for all* $x \in X$*.*

A more general definition including a set of outputs Y and an output map $H : X \to Y$ is given in [Tab09]. In our case most systems, apart from the original one, would be described with $Y = X$ and H as the identity function, thus limiting the usefulness of these elements. As stated in Section 3.1, the behavioral relationships usually relate two systems through their output behavior. Therefore, to compensate our lack of output, a map similar to H is introduced in the definition of these relationships.

Definition 3.3 (Simulation)**.** *Consider two systems $S = (X, X^0, U, \longrightarrow)$ and $S_a = (X_a, X_a^0, U_a, \underset{a}{\longrightarrow})$. A map $H : X \to X_a$ is a simulation relation from S to S_a if the following conditions hold:*

- $\forall x^0 \in X^0,\ \exists x_a^0 \in X_a^0 \mid x_a^0 = H(x^0)$,
- $\forall x \in X$, *let* $x_a = H(x) \in X_a$, *then* $\forall u \in U(x),\ \exists u_a \in U_a(x_a)$ *such that* $x' \in Post(x, u) \Rightarrow H(x') \in Post_a(x_a, u_a)$.

When H is a simulation relation from S to S_a, we say that the abstraction S_a simulates S, denoted as $S \preceq_{\mathcal{S}} S_a$.

As stated above, the map H serves as an output map for the original system S projecting its states $x \in X$ onto the state space X_a of the abstraction S_a. The first condition of Definition 3.3 requires that any initial state of S can be mapped to an initial state of the abstraction S_a: $H(X^0) \subseteq X_a^0$. The second condition means that for any transition $x \xrightarrow{u} x'$ in S, there exists an input $u_a \in U_a$ such that the transition $H(x) \underset{a}{\xrightarrow{u_a}} H(x')$ exists in S_a.

In control problems, we are interested in synthesizing a controller on the abstraction to realize some specifications and then refine it into a controller of the original system whose behavior is included in the behavior of the abstraction, thus ensuring that it also realizes the same specifications. This notion is captured in the definition of the alternating simulation.

Definition 3.4 (Alternating simulation)**.** *Consider two systems S and S_a. A map $H : X \to X_a$ is an alternating simulation relation from S_a to S if the following conditions hold:*

- $\forall x_a^0 \in X_a^0,\ \exists x^0 \in X^0 \mid x_a^0 = H(x^0)$,
- $\forall x \in X$, *let* $x_a = H(x) \in X_a$, *then* $\forall u_a \in U_a(x_a),\ \exists u \in U(x)$ *such that* $x' \in Post(x, u) \Rightarrow H(x') \in Post_a(x_a, u_a)$.

When H is an alternating simulation relation from S_a to S, we say that S alternatingly simulates the abstraction S_a, denoted as $S_a \preceq_{\mathcal{AS}} S$.

The first condition is symmetrical to the one in Definition 3.3: any initial state of the abstraction can be obtained by projecting an initial state of S onto X_a. The second condition of Definition 3.4 means that we can choose an input for the abstraction and find a corresponding input for the original system whose transitions are matched by transitions of the abstraction.

Since a second level of abstraction is introduced in Chapter 4, we need to prove the transitivity of the alternating simulation.

Proposition 3.5. *Let S_1, S_2 and S_3 be three systems such that $S_2 \preceq_{\mathcal{AS}} S_1$ and $S_3 \preceq_{\mathcal{AS}} S_2$ respectively with the alternating simulation relations $H_{12} : X_1 \to X_2$ and $H_{23} : X_2 \to X_3$. Then $H_{13} = H_{23} \circ H_{12} : X_1 \to X_3$ is an alternating simulation relation from S_3 to S_1: $S_3 \preceq_{\mathcal{AS}} S_1$.*

Proof. The first condition is immediately obtain from those of the existing alternating simulations: for all $x_3^0 \in X_3^0$, there exists $x_2^0 \in X_2^0$ such that $x_3^0 = H_{23}(x_2^0)$ and there exists $x_1^0 \in X_1^0$ such that $x_2^0 = H_{12}(x_1^0)$, which implies that $x_3^0 = H_{23}(H_{12}(x_1^0))$. For the second condition, let $x_1 \in X_1$, $x_3 = H_{23}(H_{12}(x_1)) \in X_3$ and $u_3 \in U_3(x_3)$. $S_3 \preceq_{\mathcal{AS}} S_2$ gives that there exists $u_2 \in U_2(H_{12}(x_1))$ such that for all $x_2' \in Post_2(H_{12}(x_1), u_2)$, we have $H_{23}(x_2') \in Post_3(x_3, u_3)$. Then for this particular $H_{12}(x_1) \in X_2$ and $u_2 \in U_2(H_{12}(x_1))$, $S_2 \preceq_{\mathcal{AS}} S_1$ states that there exists $u_1 \in U_1(x_1)$ such that for all $x_1' \in Post_1(x_1, u_1)$, we have $H_{12}(x_1') \in Post_2(H_{12}(x_1), u_2)$. Combining these two results, we obtain $H_{23}(H_{12}(x_1')) \in Post_3(x_3, u_3)$. □

3.2.2 Problem formulation

The continuous-time system (1.1) cannot be described as a transition system from Definition 3.1. We thus need to introduce a sampled version of (1.1) with a constant sampling period $\tau \in \mathbb{R}^+$. Let $S = (X, X^0, U, \longrightarrow)$ be this sampled system composed of:

- $X = \mathbb{R}^n$,
- $X^0 = [\underline{x}, \overline{x}) \subseteq \mathbb{R}^n$,
- $U = [\underline{u}, \overline{u}] \subseteq \mathbb{R}^p$,
- $x \xrightarrow{u} x'$ if $\exists \mathbf{w} : [0, \tau] \to [\underline{w}, \overline{w}] \mid x' = \Phi(\tau, x, u, \mathbf{w})$.

For X^0, the half-closed interval is defined similarly to (1.3): $x \in [\underline{x}, \overline{x}) \Leftrightarrow \overline{x} \gg x \succeq_x \underline{x}$. The interval is only chosen to be half-closed for practical reasons: we later want to decompose it into smaller identical intervals and to ensure that this decomposition is a partition rather than a covering, we need to use half-closed intervals. The intervals $[\underline{u}, \overline{u}]$ and $[\underline{w}, \overline{w}] \subseteq \mathbb{R}^q$ are the input bounds from Assumption 1. The transitions are defined assuming that the control input function is piecewise constant (constant between two sampling times). This assumption cannot be done on the disturbance function $\mathbf{w} : [0, \tau] \to [\underline{w}, \overline{w}]$ since we have no control over it.

Our objective is to synthesize a controller of S realizing the safety specification of maintaining its state in the interval $[\underline{x}, \overline{x})$. Since there may be more than one controller realizing this specification, we complete our specification with a performance criterion. Given a trajectory $(x^0, u^0, x^1, u^1, \dots)$ of the controlled system S, we want to minimize the performance criterion

$$\sum_{k=0}^{+\infty} \lambda^k g(x^k, u^k), \tag{3.1}$$

where $g(x, u)$ is the cost of choosing the input u when the state of S is x and $\lambda \in (0, 1)$ is a discount factor that reduces the influence of the steps further in the

future. Due to the non-determinism of the system S subject to disturbances, it is obvious that we cannot ensure the minimization of the performance criterion on all actual trajectories. Instead, in Section 3.5 and 4.4, we look at providing the tighter possible upper bound on the performance criterion $\sum_{k=0}^{+\infty} \lambda^k g(x^k, u^k)$ for any initial state $x^0 \in X^0$. These two steps for the controller synthesis problem are written formally as follows.

Control Problem 1. *Synthesize a controller $C : X \to 2^U$ such that any trajectory $(x^0, u^0, x^1, u^1, \dots)$ of the controlled system S (with $u^k \in C(x^k)$ for all $k \in \mathbb{N}$) satisfies $x^k \in [\underline{x}, \overline{x})$ for all $k \in \mathbb{N}$.*

Control Problem 2. *Refine C into a deterministic controller $C^* : X \to U$ that provides the smallest possible upper-bound of the performance criterion (3.1) for any trajectory $(x^0, u^0, x^1, u^1, \dots)$ of the controlled system S (with $u^k = C^*(x^k)$ for all $k \in \mathbb{N}$).*

3.3 Symbolic abstraction

Our objective is to synthesize a controller for S based on a symbolic abstraction of this system. To be able to do this, the symbolic abstraction of S needs to be a finite transition system. As explained in Section 3.1, such abstraction is obtained by partitioning the state space, discretizing the input set and computing a simple over-approximation of the reachable sets using the monotonicity of the system. To allow this over-approximation, Assumption 1′ is considered to be satisfied throughout this chapter: the system (1.1) is cooperative with respect to its state and disturbance, but not necessarily with respect to its control input.

State partition We start by creating a partition $\mathcal{P}^0$ of the target interval $X^0 = [\underline{x}, \overline{x}) \subseteq \mathbb{R}^n$. To take advantage of the monotonicity property satisfied by (1.1) when computing an over-approximation of the reachable sets, this interval is uniformly partitioned into smaller identical half-closed intervals. For an element $s \in \mathcal{P}^0$, we denote as $\underline{s}$ and $\overline{s}$ its lower and upper bounds, respectively: $s = [\underline{s}, \overline{s}) \subseteq \mathbb{R}^n$. If $\alpha_x \in \mathbb{N}$ denotes the number of scalar intervals per dimension of the state space, $\mathcal{P}^0$ contains α_x^n symbols and can be expressed as follows:

$$\mathcal{P}^0 = \left\{ \left[\underline{s}, \underline{s} + \frac{\overline{x} - \underline{x}}{\alpha_x} \right) \mid \underline{s} \in \left(\underline{x} + \frac{\overline{x} - \underline{x}}{\alpha_x} * \mathbb{Z}^n \right) \cap [\underline{x}, \overline{x}) \right\}, \tag{3.2}$$

where $\mathbb{Z}^n$ denotes the set of integer-valued n-dimensional vectors and $*$ is the componentwise multiplication of vectors. The partition $\mathcal{P}^0$ from (3.2) is illustrated in Figure 3.1 (a) for a 2-dimensional state space and $\alpha_x = 2$ intervals per dimension, where we can see that $(\overline{x} - \underline{x})/\alpha_x$ is the distance between the lower and upper bounds of a symbol. Although any partition into smaller intervals of various sizes is theoretically acceptable in the scope of this chapter, the choice of a uniform partition has several motivations. Firstly, it significantly simplifies the implementation task. Secondly, if the symbols are too different in sizes, the choice of the sampling period τ may become difficult. Lastly, in the compositional method in Chapter 4 we want

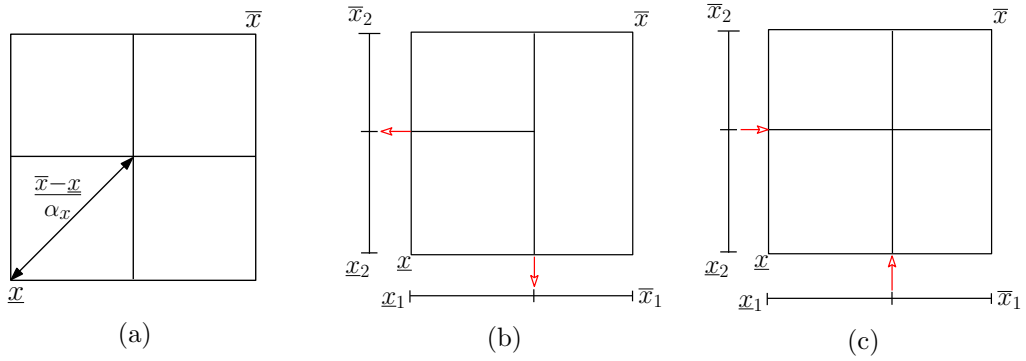


Figure 3.1 – (a) Uniform partition with $\alpha_x = 2$. (b) Projections of a non-uniform partition on each dimension. (c) Composition of the projections.

the partition of the state space to be equal to the composition of the partitions of its individual dimensions. An illustration of this last point is given in Figure 3.1, where in (b) we project the partition on each dimension and in (c) we can see that the composition of these projections does not lead to the same partition. A partition $\mathcal{P}$ of the whole state space $X = \mathbb{R}^n$ is then obtained by adding the symbol $Out = \mathbb{R}^n \backslash [\underline{x}, \overline{x})$ to the partition $\mathcal{P}^0$:

$$\mathcal{P} = \mathcal{P}^0 \cup Out. \tag{3.3}$$

Input discretization The next step is to discretize the input set $U = [\underline{u}, \overline{u}] \subseteq \mathbb{R}^p$. Similarly to how the symbol lower bounds $\underline{s}$ are obtained in (3.2), we uniformly discretize $[\underline{u}, \overline{u}]$ into $\alpha_u \in \mathbb{N}$ values per dimension. The difference with (3.2) is that we impose $\alpha_u \geq 2$ to ensure that our discrete input set U^d contains at least both values $\underline{u}$ and $\overline{u}$:

$$U^d = \left(\underline{u} + \frac{\overline{u} - \underline{u}}{\alpha_u - 1} * \mathbb{Z}^p \right) \cap [\underline{u}, \overline{u}]. \tag{3.4}$$

Transitions With Assumption 1′, the system (1.8) with an time-dependent vector field $F(t, x, w) = f(x, u(t), w)$ is cooperative. We can then use Definition 1.10 with a constant control input u over the sampling period to compute an over-approximation of the reachable set $\bigcup_{x \in [\underline{s}, \overline{s})} Post(x, u)$ of S from all continuous states in a symbol $s = [\underline{s}, \overline{s})$:

$$\forall x \in s,\ \mathbf{w} : [0, \tau] \to [\underline{w}, \overline{w}],\ \Phi(\tau, x, u, \mathbf{w}) \in [\Phi(\tau, \underline{s}, u, \underline{w}), \Phi(\tau, \overline{s}, u, \overline{w})]. \tag{3.5}$$

The symbolic abstraction can thus be defined as $S_a = (X_a, X_a^0, U_a, \underset{a}{\longrightarrow})$ with:

- $X_a^0 = \mathcal{P}^0$ as in (3.2),
- $X_a = \mathcal{P}$ as in (3.3),
- $U_a = U^d$ as in (3.4),

- $\forall s \in \mathcal{P}^0,\ u \in U^d,\ s' \in \mathcal{P}$,
  $s \xrightarrow[a]{u} s' \iff s' \cap [\Phi(\tau, \underline{s}, u, \underline{w}), \Phi(\tau, \overline{s}, u, \overline{w})] \neq \emptyset$,
- $\forall u \in U^d,\ s' \in \mathcal{P},\ Out \xrightarrow[a]{u} s'$.

According to the fourth point, for a symbol $s \in \mathcal{P}^0$ and an input $u \in U^d$, the successors in S_a are the symbols partially covered by the over-approximation of the reachable set (3.5). The last point completes the definition of the transitions. Since the symbol Out is not bounded, (3.5) cannot be applied. To avoid a complex and costly consideration on the continuous states of S that can be reached from a state in $Out = \mathbb{R}^n \backslash [\underline{x}, \overline{x})$, we simply consider that all transitions are possibles: for all $u \in U^d$, $Post_a(Out, u) = \mathcal{P}$. Apart from its simplicity, the second reason for this choice is to ensure the alternating simulation between S_a and S: we need to make sure that all transitions of S from a state in $\mathbb{R}^n \backslash [\underline{x}, \overline{x})$ have a match in S_a from Out. This over-approximation of the reachable set $Post(Out, u)$ has no consequence in what follows as the realization of the safety specification will discard all these transitions from the unsafe symbol Out.

Proposition 3.6. *Under Assumption 1′, the map $H_a : \mathbb{R}^n \to \mathcal{P}$ defined by*

$$s = H_a(x) \Leftrightarrow x \in s$$

is an alternating simulation relation from S_a to S: $S_a \preceq_{\mathcal{AS}} S$.

Proof. The first condition of Definition 3.4 is immediately satisfied since $\mathcal{P}^0$ is a partition of $[\underline{x}, \overline{x})$. For the second condition, let $s = [\underline{s}, \overline{s}) \in \mathcal{P}^0$, $x \in s$, $u \in U_a(s) = U_a \subseteq U = U(x)$ (Remark 3.2) and $x' \in Post(x, u)$. From the definition of the transitions of S and (3.5) that exploits the cooperativeness of the continuous system (Assumption 1′), $Post(x, u) \subseteq [\Phi(\tau, \underline{s}, u, \underline{w}), \Phi(\tau, \overline{s}, u, \overline{w})]$ which means that $H_a(x') \in Post_a(s, u)$. Lastly, $\forall u \in U_a,\ Post_a(Out, u) = \mathcal{P}$, therefore any transition in S from $x \in Out$ can be matched by a transition in S_a. □

As indented, the symbolic model S_a described above is a finite-state and finite-transition abstraction of the initial system S. In addition, for a pair (s, u), checking the existing outgoing transitions $s \xrightarrow[a]{u} s'$ only requires to compute two successors in S (the bounds of s) and intersect the obtained over-approximation interval with the finite partition $\mathcal{P}$. This symbolic model can thus be built with a finite number of operations.

3.4 Abstraction-based controller synthesis

In this section, we use the finite symbolic abstraction of S to synthesize a controller realizing the specifications given in Section 3.2.2. This is done in two steps. Firstly, a non-deterministic controller realizing the safety specification is synthesized with a classical fixed-point algorithm. Then we choose among the safe control values using a receding horizon control scheme on the result of the optimization of an approximation of the performance criterion (3.1).

3.4.1 Safety controller synthesis

In Control Problem 1 formulated on S in Section 3.2.2, the safety objective is to find a controller $C : X \to 2^U$ such that any trajectory $(x^0, u^0, x^1, u^1, \dots)$ of system S controlled with C ($u^k \in C(x^k)$ for all $k \in \mathbb{N}$) satisfies $x^k \in [\underline{x}, \overline{x})$ for all $k \in \mathbb{N}$. A similar safety specification can be expressed on the symbolic abstraction: we want to synthesize a controller $C_a : X_a \to 2^{U_a}$ such that any trajectory $(s^0, u^0, s^1, u^1, \dots)$ of system S_a controlled with C_a ($u^k \in C_a(s^k)$ for all $k \in \mathbb{N}$) satisfies $s^k \in \mathcal{P}^0$ for all $k \in \mathbb{N}$. This safety game on S_a can be solved by introducing the operator $F_{\mathcal{P}^0} : 2^{\mathcal{P}} \to 2^{\mathcal{P}}$ such that:

$$F_{\mathcal{P}^0}(Z) = \{s \in Z \cap \mathcal{P}^0 \mid \exists u \in U_a,\ Post_a(s, u) \subseteq Z\}, \tag{3.6}$$

where the set $F_{\mathcal{P}^0}(Z)$ contains all symbols $s \in Z \cap \mathcal{P}^0$ whose successors stay in Z for some $u \in U_a$. Note that from Remark 3.2 we have $U_a(s) = U_a$ for all $s \in \mathcal{P}^0$, which thus implies that $Post_a(s, u) \neq \emptyset$ in (3.6). Since the symbolic abstraction S_a is a finite transition system, the maximal fixed-point $Z_a = \lim_{k \to \infty} F^k_{\mathcal{P}^0}(\mathcal{P}^0)$ of $F_{\mathcal{P}^0}$ can be obtained in a finite number of steps. This fixed point $Z_a \subseteq \mathcal{P}^0$ thus corresponds to the maximal *safe set* for S_a: for any symbol in Z_a, we can find a control input such that all successors stay in Z_a. It also allows the definition of a non-deterministic controller $C_a : Z_a \to 2^{U_a}$ solving the safety game for S_a if $Z_a \neq \emptyset$ [Tab09]:

$$C_a(s) = \{u \in U_a \mid Post_a(s, u) \subseteq Z_a\}. \tag{3.7}$$

Let Z_a^X be defined as the union in $X = \mathbb{R}^n$ of all the safe symbols in Z_a:

$$Z_a^X = \{x \in \mathbb{R}^n \mid \exists s \in Z_a,\ x \in s\}. \tag{3.8}$$

With the alternating simulation relation H_a in Proposition 3.6, we can refine C_a into a controller $C_a^X : Z_a^X \to 2^U$ of the sampled system S:

$$\forall x \in Z_a^X,\ C_a^X(x) = C_a(H_a(x)). \tag{3.9}$$

We can then prove that C_a^X is a safety controller solving Control Problem 1 for S.

Theorem 3.7. *$Z_a^X \subseteq [\underline{x}, \overline{x})$ is a safe set for system S controlled with any strategy of C_a^X.*

Proof. Let $x \in Z_a^X$, $u \in C_a^X(x) = C_a(H_a(x))$ and $x' \in Post(x, u)$. Combining the second condition of the alternating simulation (Definition 3.4) and the definition of C_a (3.7), we obtain $H_a(x') \in Post_a(H_a(x), u) \subseteq Z_a$ which implies that $x' \in Z_a^X$. □

Example 3.1. We can illustrate the synthesis of C_a on the temperature diffusion example (1.10) from Section 1.3.1 in the conditions of Example 2.2, where the chosen target interval with $\underline{x} = (22; 18)$ and $\overline{x} = (24; 23)$ is robust controlled invariant. We create the symbolic abstraction S_a with the sampling period $\tau = 0.1$ and the parameters $\alpha_x = 2$ and $\alpha_u = 3$. Since the state space is $\mathbb{R}^2$ and we have a single control input, this means that S_a has $\alpha_x^2 = 4$ symbols and $\alpha_u = 3$ discrete control values. The obtained transition system S_a is given in Figure 3.2 where a color is

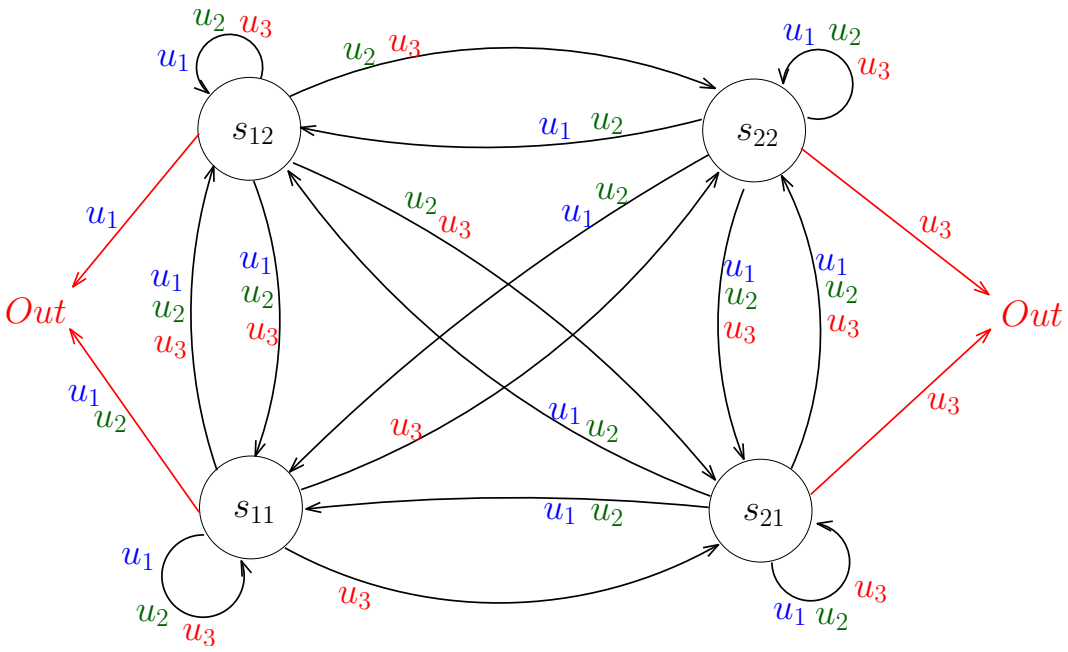


Figure 3.2 – Symbolic abstraction S_a for the diffusion system with $\alpha_x = 2$ and $\alpha_u = 3$.

affected to each of the three control values for a better visualization: $u_1 = \underline{u} = 18\,°\text{C}$ is the coldest control, $u_2 = 24\,°\text{C}$ is the central value and $u_3 = \overline{u} = 30\,°\text{C}$ is the hottest control.

We can see in Figure 3.2 that all symbols have some transitions going to the unsafe symbol *Out*. If we apply the operator (3.6) to the partition of the interval, we reach a fixed point in a single step: $F_{\mathcal{P}^0}(\mathcal{P}^0) = \mathcal{P}^0$. We can indeed see that each symbol has at least a control value whose successors all are in $\mathcal{P}^0$. The safety controller C_a from (3.7) can then be deduced by forbidding the inputs that may lead to the symbol *Out*: u_1 cannot be used from s_{11} and s_{12}, u_2 from s_{11} and u_3 from s_{21} and s_{22}. Applying this safety controller to S_a, we obtain the transition system in Figure 3.3 where we can observe that the unsafe symbol *Out* is not reachable anymore.

Obtaining a safe set containing all the symbols of the partitions ($Z_a = \mathcal{P}^0$) could have been expected in these conditions. A safe set for S can be assimilated to the notion of robust controlled invariant set for a discrete-time system and we know from Example 2.2 that the target interval $[\underline{x}, \overline{x})$ used for this example is robust controlled invariant for the continuous-time system (1.1) as in Definition 2.4. Then as long as the sampling period τ does not take too large values, it is natural that the safe set Z_a^X covers the whole interval. △

The above example illustrates the main idea behind the safety controller synthesis in a very simple case where the fixed-point of $F_{\mathcal{P}^0}$ is reached after a single step of the operator in (3.6), thus leading to a safe set $Z_a = \mathcal{P}^0$. In the next example, we provide some cases where the safe set does not cover the whole partition of the interval. In particular, we compare these results with the notion of robust controlled

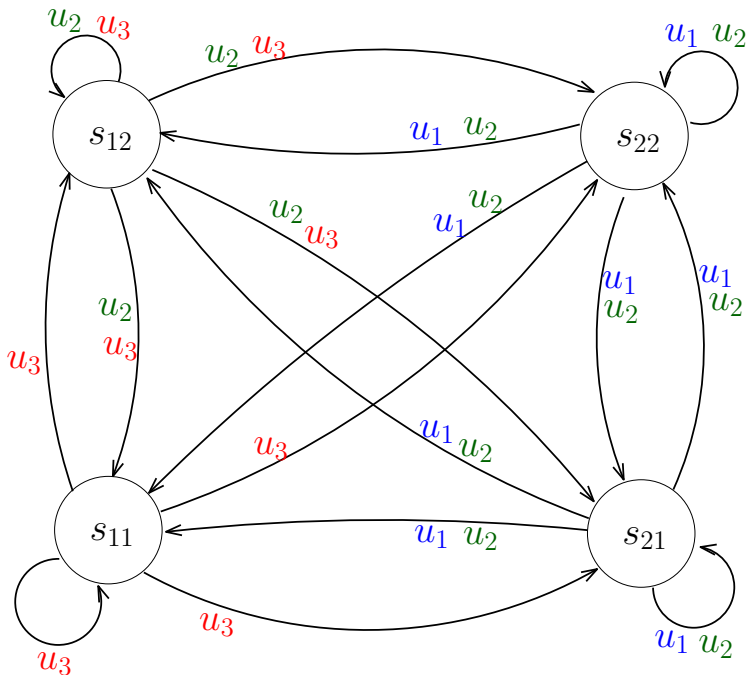


Figure 3.3 – Symbolic abstraction S_a constrained by the safety controller C_a.

invariant interval introduced in Chapter 2.

Example 3.2. We consider the coupled-tank example from Section 1.3.2 with the safety specifications $x_2 \in [15, 20)$ for tank 2 and almost no constraint on the water level in tank 1 ($x_1 \in [1, 30)$). In the top graph of Figure 3.4, the thin black lines represent the target interval and its partition $\mathcal{P}^0$ into symbols. The gray area corresponds to the union Z_a^X of all the safe symbols in the safe set Z_a. The blue and red sets in the background are the sets described in Figure 2.4 and Example 2.4 where the lower and upper bounds of an interval need to be chosen to satisfy the robust controlled invariance from Theorem 2.5. The boundaries of these sets are drawn in thicker blue and red lines and are used to find the maximal robust controlled invariant sub-interval, that is the largest interval (in the sense of set inclusion) which satisfies Theorem 2.5 and is contained in the target interval $[\underline{x}, \overline{x})$. This interval is obtained by lowering $\overline{x}_1$ until it reaches the red set and increasing $\underline{x}_1$ until it reaches the blue set. The other two graphs in Figure 3.4 only represent the target interval, its safe set Z_a^X and the maximal robust controlled invariant sub-interval.

Before comparing the safe set Z_a^X with the robust controlled invariance, let us discuss the influence of the sampling period τ on the quality of the safety results. For three different sampling, we create the symbolic abstraction with $\alpha_x = 10$ symbols and $\alpha_u = 4$ control values per dimension and synthesize a safety controller realizing the same specifications described at the beginning of this example. In Table 3.1, we give for each sampling value the number of iteration of the operator $F_{\mathcal{P}^0}$ in (3.6) before reaching the maximal fixed-point Z_a and the number of symbols contained in this safe set. Among these three values, the ideal choice is $\tau = 0.5\,\mathrm{s}$ which corresponds to the top graph of Figure 3.4. When we increase the sampling period,

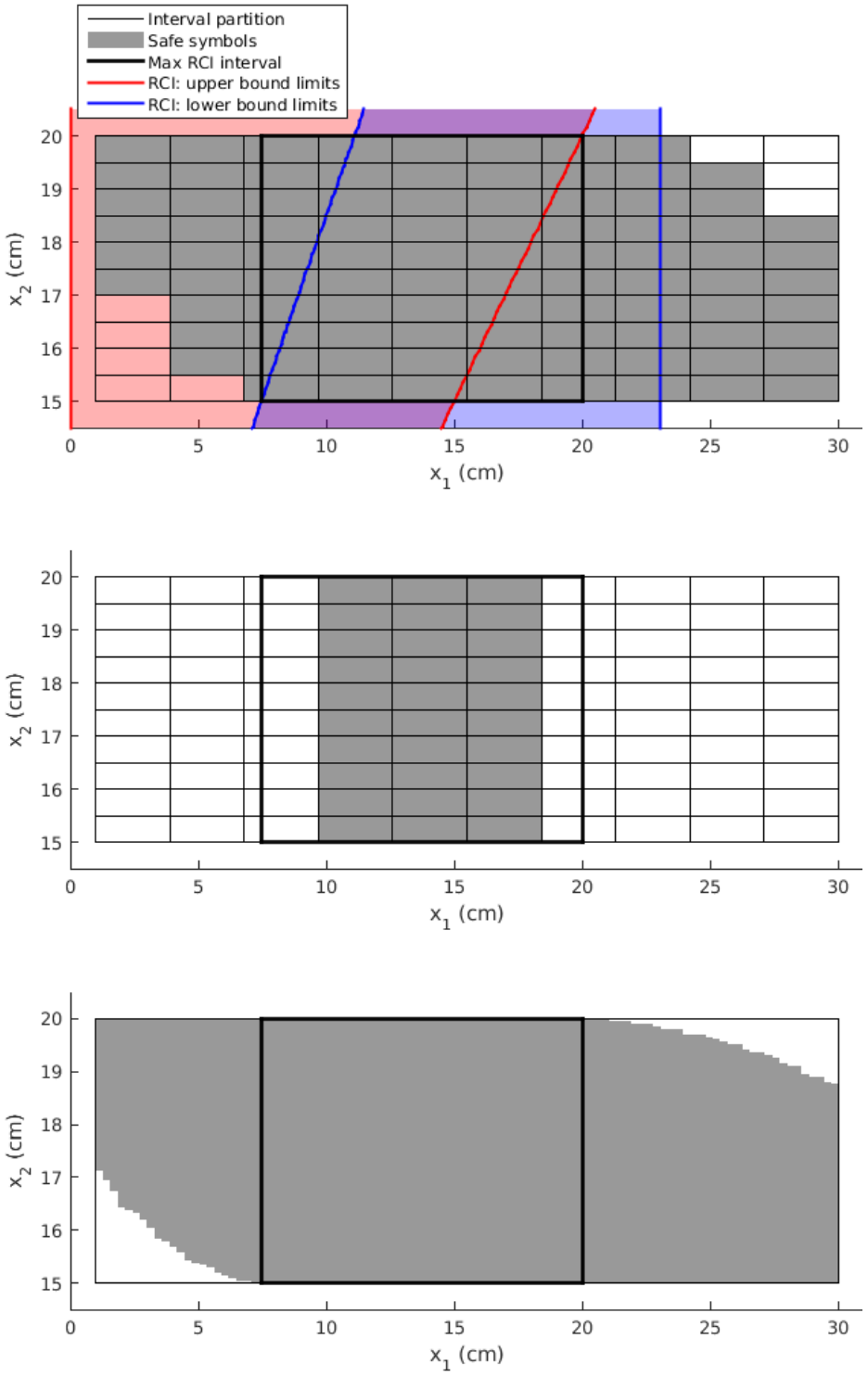


Figure 3.4 – Interval partition with safe symbols (gray), robust controlled invariance limits from Figure 2.4 (red and blue sets) and maximal robust controlled invariant sub-interval (thick black). Top: $\alpha_x = 10$ and $\tau = 0.5\,\mathrm{s}$. Center: $\alpha_x = 10$ and $\tau = 0.1\,\mathrm{s}$. Bottom: $\alpha_x = 100$ and $\tau = 0.1\,\mathrm{s}$.

the reachable set from a symbol goes too far, either outside of the safety specification of to other symbols that are unsafe. This can be seen in Table 3.1 for $\tau = 1\,\mathrm{s}$, where all symbols are detected as unsafe only after two iterations of $F_{\mathcal{P}^0}$, the third iteration simply confirming that a fixed-point (empty in this case) was reached. On the other hand, with a too low sampling period we may obtain less interesting safety results as in the central graph in Figure 3.4 corresponding to $\tau = 0.1\,\mathrm{s}$. Indeed, with a small sampling period, the over-approximation of the reachable set from a given symbol does not go far enough from this symbol and thus necessarily intersects some of its immediate neighbors. For example, if we consider the unsafe symbols in the top-right corner of the top graph in Figure 3.4, at each iteration of $F_{\mathcal{P}^0}$ the symbol below reaches the unsafe symbol, which then propagates until the whole column is unsafe. This is why in the central graph the safe set Z_a^X is rectangular.

Sampling period τ	0.1 s	0.5 s	1 s
Number of iterations of $F_{\mathcal{P}^0}$	10	2	3
Safe symbols in Z_a (max. 100)	30	91	0

Table 3.1 – Number of iteration of the operator (3.6) before reaching a fixed-point Z_a and number of safe symbols in the fixed-point, for three values of the sampling period τ with $\alpha_x = 10$ and $\alpha_u = 4$.

In the case where $\tau = 0.5\,\mathrm{s}$ (top graph of Figure 3.4), we can see that the safe set Z_a^X (gray symbols) contains the largest robust controlled invariant interval that can be found inside the target interval $[\underline{x}, \overline{x})$. This is always the case as long as the partition is not too coarse and the sampling period is chosen correctly. Note that the safe set Z_a^X corresponds to a robust controlled invariant set for the discrete-time system S and not necessarily for the continuous-time dynamics (1.1). It really is comparable with the robust controlled invariant interval only when the precision α_x of the partition grows and the sampling period τ decreases accordingly. In the bottom graph of Figure 3.4, we solve the same safety specifications with $\alpha_x = 100$ symbols per dimensions and $\tau = 0.1\,\mathrm{s}$. There, we can see that the maximal robust controlled invariant *sub-interval* of $[\underline{x}, \overline{x}]$ (thick black lines) is much smaller than the safe set Z_a^X which approaches the maximal robust controlled invariant *subset* of $[\underline{x}, \overline{x})$ (gray set) for the continuous-time system (1.1). △

Some guidelines on the choice of the sampling period τ depending on the precision α_x of the state space partition can be found in [SP94]. In this paper, an autonomous system is considered and the viability kernel (corresponding to the maximal invariant set in our scope) is approximated using discrete viability kernels of sampled versions of the system with quantized state. It is then proven that the discrete viability kernels converge to the continuous viability kernel when the sampling and quantization steps go to zero while satisfying some condition. This condition, adapted to our method, is written as:

$$2L\tau^2 \sup_{x \in [\underline{x}, \overline{x})} \|f(x, \overline{u}, \overline{w})\| \geq \frac{\|\overline{x} - \underline{x}\|}{\alpha_x}.$$

This links the sampling step τ with the partition step $\|\overline{x} - \underline{x}\|/\alpha_x$ and involves the

Lipschitz constant L and the supremum of the vector field in the considered interval $[\underline{x}, \overline{x})$.

3.4.2 Receding horizon control

Performance optimization The safety controller C_a defined in (3.7) is non-deterministic as a symbol $s \in Z_a$ may use several safe control actions. We thus want to choose the best control input for each symbol according to an approximation of the performance criterion (3.1). In Control Problem 2, the criterion of interest is given by $\sum_{k=0}^{+\infty} \lambda^k g(x^k, u^k)$ where $g(x^k, u^k) \in \mathbb{R}^+$ is the cost of choosing input u^k when the state of S at the k^{th} time step is x^k. Since we use the symbolic abstraction S_a, the actual states of S are not available and the values of function g cannot be computed. We thus need to work with a new cost function for S_a defined as follows:

$$g_a(s, u) = \max_{x \in s} g(x, u). \tag{3.10}$$

Following a similar strategy as for the over-approximation of the reachable set (3.5), the cost $g_a(s, u)$ is taken as the worst case of the costs $g(x, u)$ for $x \in s$. Since the total cost $\sum_{k=0}^{+\infty} \lambda^k g_a(s^k, u^k)$ on an infinite trajectory $(s^0, u^0, s^1, u^1, \dots)$ cannot be computed in a finite number of iteration, it is approximated by the new performance criterion

$$\sum_{k=0}^{N} \lambda^k g_a(s^k, u^k) \tag{3.11}$$

on a finite horizon of $N \in \mathbb{N}$ sampling periods. This approximation is reasonable if N and the discount factor $\lambda \in (0, 1)$ are chosen such that λ^{N+1} is sufficiently small and the remaining cost of the trajectory can be neglected: $\sum_{k=N+1}^{+\infty} \lambda^k g_a(s^k, u^k) \approx 0$.

The symbolic abstraction is non-deterministic and (3.11) cannot be directly computed since we do not know in advance which successor $s^{k+1} \in Post(s^k, u^k)$ will appear. We thus use a dynamic programming algorithm [Ber95] to minimize a (3.11) using worst-case predictions of the future steps. For any initial state s^0, we define the cost $J_a^0(s^0)$ that is computed iteratively following the principle of optimality:

$$J_a^N(s) = \min_{u \in C_a(s)} g_a(s, u), \tag{3.12a}$$

$$J_a^k(s) = \min_{u \in C_a(s)} \left(g_a(s, u) + \lambda \max_{s' \in Post_a(s,u)} J_a^{k+1}(s') \right). \tag{3.12b}$$

At each step k from N to 0 in (3.12), we minimize over the safe inputs $u \in C_a(s)$ the sum of the cost of the current step and the worst-case additive cost of all the following steps. The first step (3.12a) of the algorithm ($k = N$) only minimizes the cost $g_a(s, u)$ since the additive cost of the following steps J_a^{N+1} is neglected. Here we consider worst-case predictions because we take a robust approach, but if we have probabilistic distributions of the causes of the non-determinism we can replace the *max* operator by the expectation. The discount factor $\lambda \in (0, 1)$ is used to reduce the influence of the cost of future steps significantly affected by the non-determinism of S_a. The result of (3.12) is a control policy $(u^0(s), \dots, u^N(s))$ for each initial symbol $s \in Z_a$. We should note that this policy is only optimal in the

conditions of (3.12) where we take the worst-case prediction of the future steps and it may not be the best control strategy to minimize (3.11) on an actual trajectory of S_a when the disturbance does not lead to the worst case of non-determinism.

Remark 3.8. *Some cost function may need to involve not only the symbol and the control action of the current step, but also memories of their previous values. In that case, the cost functions g_a and $\hat{g}_a$ in (3.10) and the additive costs J_a^k in (3.12) need to be redefined by replacing the current symbol s^k by an extended state $z^k = (s^k, u^{k-1}, s^{k-1}, \dots)$ containing the current symbol and the memory of the previous symbols and inputs of the trajectory [Ber95]. Although it allows the consideration of a wider variety of performance criteria, it also significantly increases the computational cost.*

Receding horizon We can then apply a receding horizon control scheme where we measure the current symbol s and only apply the first element $u^0(s)$ of the control policy provided by (3.12), then repeat at the next sampling time. The obtained controller can be described by (3.12) where the last iteration ($k = 0$) is replaced by the following:

$$C_a^*(s) = \underset{u \in C_a(s)}{\arg\min} \left(g_a(s,u) + \lambda \max_{s' \in Post_a(s,u)} J_a^1(s') \right). \tag{3.13}$$

This approach is the basis of model predictive control [RM09], with the difference that all the computations of (3.12) and (3.13) can be done offline for our finite transition system S_a. This method is also used in [DLB14] for the control of a deterministic finite-state system to satisfy temporal logic formula. In our case, the system S_a is non-deterministic due to both the disturbance and the abstraction done in Section 3.3. With the alternating simulation relation H_a in Proposition 3.6 and the set Z_a^X in (3.8), we can refine C_a^* into a controller $C_a^{*X} : [\underline{x}, \overline{x}) \to U$ of the sampled system S:

$$\forall x \in Z_a^X,\ C_a^{*X}(x) = C_a^*(H_a(x)). \tag{3.14}$$

We can apply this optimization in the simple case illustrated in Example 3.1.

Example 3.3. In Example 3.1, we considered the symbolic abstraction for the temperature diffusion system (1.10). The result of the safety controller synthesis was the following non-deterministic controller C_a:

$$C_a(s_{11}) = \{u_3\},\ C_a(s_{12}) = \{u_2, u_3\},\ C_a(s_{21}) = \{u_1, u_2\},\ C_a(s_{22}) = \{u_1, u_2\}.$$

To refine C_a into a deterministic controller C_a^*, we take the cost function $g(x,u) = u$. Our goal is thus to minimize the value of the control input u. For this particularly small example, the choice of the horizon size N and the discount factor λ has no influence over the final result. As it can be seen in Figure 3.3, this is due to the fact that for any symbol s and any safe control $u \in C_a(s)$, the set of successors contains the whole partition: $Post_a(s,u) = \mathcal{P}^0$. This implies that $\lambda \max_{s' \in Post_a(s,u)} J_a^{k+1}(s')$ is independent of s and u, which means that at each step k, we have

$$J_a^k(s) = \min_{u \in C_a(s)} (g_a(s,u)) + \lambda \max_{s' \in \mathcal{P}^0} J_a^{k+1}(s').$$

Then, for any value of N, we simply minimize $g(x,u) = u$ over the safe control inputs $u \in C_a(s)$. The deterministic controller is thus given by:

$$C_a^*(s_{11}) = u_3,\ C_a^*(s_{12}) = u_2,\ C_a^*(s_{21}) = u_1,\ C_a^*(s_{22}) = u_1. \qquad \triangle$$

3.5 Performance guarantee

In this section, to solve Control Problem 2, we provide some guarantees on the performance criterion (3.1) for any trajectory $(x^0, u^0, x^1, u^1, \dots)$ of the system S controlled with the deterministic controller C_a^{*X} from (3.14). Let $M_a \in \mathbb{R}^+$ denote the maximal value of the first step ($k = N$) of the dynamic programming algorithm (3.12a) over the safe symbols:

$$M_a = \max_{s \in Z_a} J_a^N(s) = \max_{s \in Z_a} \min_{u \in C_a(s)} g_a(s,u). \tag{3.15}$$

This upper bound M_a of J_a^N is used in the following intermediate result.

Lemma 3.9. *$J_a^0(s) \leq J_a^1(s) + \lambda^N M_a$ for all $s \in Z_a$.*

Proof. This is proved by induction. For the initial step, we consider the second part of the dynamic programming algorithm (3.12b) with $k = N-1$ and the input $u \in C_a(s)$ satisfying $J_a^N(s) = g_a(s,u)$ in (3.12a), then use (3.15):

$$J_a^{N-1}(s) \leq J_a^N(s) + \lambda \max_{s' \in Post_a(s,u)} J_a^N(s') \leq J_a^N(s) + \lambda M_a.$$

Assume now that $J_a^k(s) \leq J_a^{k+1}(s) + \lambda^{N-k} M_a$, then:

$$\begin{aligned} J_a^{k-1}(s) =& \min_{u \in C_a(s)} \left(g_a(s,u) + \lambda \max_{s' \in Post_a(s,u)} J_a^k(s') \right) \\ \leq& \min_{u \in C_a(s)} \left(g_a(s,u) + \lambda \max_{s' \in Post_a(s,u)} J_a^{k+1}(s') \right) + \lambda^{N-k+1} M_a \\ \leq& J_a^k(s) + \lambda^{N-k+1} M_a. \end{aligned}$$

With $k = 1$, we obtain the result in Lemma 3.9. □

For any trajectory of the controlled system, we can then obtain an upper bound of the performance criterion (3.1) starting on any state of the trajectory.

Theorem 3.10. *Let $(x^0, u^0, x^1, u^1, \dots)$ with $x^0 \in Z_a^X$ be a trajectory of S controlled with C_a^{*X} in (3.14). Then for all $k \in \mathbb{N}$,*

$$\sum_{j=0}^{+\infty} \lambda^j g(x^{k+j}, u^{k+j}) \leq J_a^0(H_a(x^k)) + \frac{\lambda^{N+1}}{1-\lambda} M_a.$$

Proof. Combining (3.14), (3.13) and (3.9), we have $C_a^{*X}(x) \in C_a^X(x)$ for all $x \in Z_a^X$. Then with Theorem 3.7, we know that C_a^{*X} also is a safety controller for S, which implies $x^k \in Z_a^X$ for all $k \geq 1$ if $x^0 \in Z_a^X$. To simplify the notations, let

$s^k = H_a(x^k) \in \mathcal{P}^0$ for all $k \in \mathbb{N}$ and $J_a(s) = J_a^0(s) + \frac{\lambda^{N+1}}{1-\lambda} M_a$. We start from the definition of $J_a^0(s^k)$ in (3.12) with $u^k = C_a^*(s^k)$ as in (3.13):

$$\begin{aligned} J_a(s^k) &= g_a(s^k, u^k) + \lambda \max_{s' \in Post_a(s^k, u^k)} J_a^1(s') + \frac{\lambda^{N+1}}{1-\lambda} M_a \\ &\geq g_a(s^k, u^k) + \lambda J_a^1(s^{k+1}) + \frac{\lambda^{N+1}}{1-\lambda} M_a \\ &\geq g_a(s^k, u^k) + \lambda \left(J_a^0(s^{k+1}) - \lambda^N M_a + \frac{\lambda^N}{1-\lambda} M_a \right) \\ &\geq g(x^k, u^k) + \lambda J_a(s^{k+1}) \end{aligned}$$

The first inequality is obtained for a particular value $s' = s^{k+1}$ of the possible successors, the second comes from Lemma 3.9 and the third from the definition (3.10) of g_a. Thus, if the inequality obtained above is applied to all the following states of the trajectory, we have for any k:

$$\begin{aligned} J_a(s^k) &\geq g(x^k, u^k) + \lambda J_a(s^{k+1}) \\ &\geq g(x^k, u^k) + \lambda g(x^{k+1}, u^{k+1}) + \lambda^2 J_a(s^{k+2}) \\ &\geq \dots \end{aligned}$$

Expanding these inequalities to all states of the trajectory leads to the result in Theorem 3.10. □

The upper bound in Theorem 3.10 contains two elements. $J_a^0(H_a(x^k))$ is the worst-case minimization of the performance criterion (3.11) on S_a for the finite horizon of N sampling periods, which is naturally greater than the real cost on S restricted to the finite horizon: $\sum_{j=0}^{N} \lambda^j g(x^{k+j}, u^{k+j})$. Since on state x^k the optimization only runs until the time $k+N$, the only available information on the rest of the infinite trajectory is that the receding horizon method will at least minimize the costs $g_a(s^{k+j}, u^{k+j})$ when it reaches the time $k+j$. As the state x^{k+j} and the symbol s^{k+j} are unknown, we need to take the worst-case of this minimization:

$$\sum_{j=N+1}^{+\infty} \lambda^j g(x^{k+j}, u^{k+j}) \leq \sum_{j=N+1}^{+\infty} \lambda^j g_a(s^{k+j}, u^{k+j}) \leq \sum_{j=N+1}^{+\infty} \lambda^j \max_{s \in Z_a} \min_{u \in C_a(s)} g_a(s, u),$$

resulting in the constant part $\frac{\lambda^{N+1}}{1-\lambda} M_a$ of the upper bound, with M_a defined in (3.15). Note that this term goes to zero when the size N of the horizon used in the dynamic programming grows.

Chapter 4

Compositional approach to symbolic control

The centralized approach for symbolic control presented in Chapter 3 suffers from a scalability issue since its complexity grows exponentially in the dimension of the state and input spaces. In this chapter, we study a compositional solution where the control of the whole system is deduced from reasoning on subsystems partially describing the global behavior. The symbolic methods from Chapter 3 can be applied to each subsystem with a reduced complexity, but at the cost of a more conservative approach since all variables that are not observed in a subsystem have to be considered as external disturbances. We thus take a tradeoff between a reduced complexity and the precision of the model.

This chapter is organized as follows. We first motivate this compositional approach and review the related literature as well as other solutions to solve the scalability issue in Section 4.1. Then, in Section 4.2, we introduce some notations to describe the decomposition of the global dynamics into subsystems. In Section 4.3, we create the symbolic abstractions of these subsystems and synthesize the associated controllers as in Chapter 3. Then in Section 4.4, we prove that the safety and performance guarantees can be realized with this approach. The performance guarantee and the complexity of this compositional approach are compared to those of the centralized method from Chapter 3 in Sections 4.5 and 4.6, respectively. Finally in Section 4.7, we describe some particular cases of the decomposition into subsystems.

This compositional approach is presented in [MGWb] for a less general case where all observed states also are controlled.

4.1 Motivations and related work

Scalability One of the main challenges affecting the symbolic control methods presented in Chapter 3 is scalability. The computational cost of the controller synthesis from a symbolic abstraction (described as a finite transition system) mainly depends on four elements: the number of symbols, the number of input values, the number of transitions for each pair symbol-input (influenced by the non-determinism) and the

complexity of the specifications. Since symbolic abstractions are usually obtained based on partition or quantization of the state space, the number of symbols tends to grow exponentially with the dimension of the state space. The number of discrete inputs also grows exponentially with the dimension of the input space when they are obtained from a discretization of a continuous input set. As a result, symbolic control methods are limited to systems with relatively low dimensions. An approach to this problem is to look for an intermediate continuous abstraction of the system with a lower dimension [GP09, MR02] and only then create a discrete abstraction of the reduced model. When the input space is of lower dimension than the state space, it can be interesting to use the method presented in [Gir14] where no partition of the state space is required and a partition of the input space is used so that the symbols of the abstraction are sets of continuous states that can be reached from a sequence of inputs. Another approach is to consider an abstraction with several levels of precision not computed beforehand: when it is sufficient we work on the coarser level, then the finer levels of abstraction are only computed on the fly when needed [CGG11].

Compositional reasoning In this chapter, our main objective is to solve this scalability issue with a compositional approach. The motivation for this approach is linked to the complexity of verification or control problem on discrete systems. As stated above, since the complexity is exponential in the dimensions of the state and input spaces, instead of working on a high-dimensional model describing the whole dynamics of the system, we decompose the system into subsystems of lower dimensions. Each of these subsystems partially describes the global model by only focusing on a subset of the state and input components. The verification or control synthesis tasks are then achieved on each subsystem at a significantly lower computational cost. Assume that we want to verify a property Q on a system S, denoted as $S \to Q$. Consider that S can be decomposed into two subsystems S_1 and S_2 ($S = S_1 \| S_2$, where $\|$ denotes some composition operator) and similarly, Q can be written as $Q = Q_1 \| Q_2$. Then, the principle of the compositional method is that S satisfies Q if S_1 and S_2 satisfy Q_1 and Q_2, respectively:

$$\begin{cases} S_1 \to Q_1 \\ S_2 \to Q_2 \end{cases} \Rightarrow S_1 \| S_2 \to Q_1 \| Q_2.$$

An overview and survey on compositional reasoning can be found in [dR98].

Assume-Guarantee While such method usually is sound, it is often too restrictive to solely look at the behavior of a subsystem without consideration on the others: the satisfaction of the desired properties on the global system usually comes from the interconnection of its components. This leads to a new type of compositional approaches where the verification or synthesis tasks are not applied to the subsystem but to the subsystem constrained by its environment representing the interconnections with other components. This method was independently introduced in [Jon83] as *rely-guarantee* and in [MC81] as *assumption-commitment*. The name of *assume-guarantee* reasoning later established itself, mainly influenced by the work of

Henzinger and co-authors who worked, among other things, on verification [HQR98], controller synthesis [CH07] or checking simulation relations [HQRT98] and mainly focused on systems formalized as reactive modules [AH99]. Taking back the above example, a possible assume-guarantee reasoning would be to look for the property Q_1 on S_1 when Q_2 is assumed to be satisfied and symmetrically, look for Q_2 on S_2 constrained by Q_1. We thus want an implication as follows:

$$\begin{cases} S_1 \| Q_2 \to Q_1 \\ Q_1 \| S_2 \to Q_2 \end{cases} \Rightarrow S_1 \| S_2 \to Q_1 \| Q_2.$$

Note that there may be different formulations, e.g. depending on the composition operator $\|$. This type of reasoning is interesting only when we can prove that this implication is true. In particular, the circular dependence on $S_1 \| Q_2 \to Q_1$ and $Q_1 \| S_2 \to Q_2$ may require an additional condition to break this circularity (e.g. see [VV01]).

Symbolic composition In the scope of abstraction-based methods, most compositional approaches in the literature have a similar goal: to prove that the simulation relation (or its variants) is preserved under composition. To illustrate the basic idea of this approach, let S_1, S_2, Σ_1, Σ_2 be four systems and $\preceq$ represent a behavioral relationship as described above, such as a simulation relation. Then, the goal is to prove that we have the following implication

$$\begin{cases} S_1 \preceq \Sigma_1 \\ S_2 \preceq \Sigma_2 \end{cases} \Rightarrow S_1 \| S_2 \preceq \Sigma_1 \| \Sigma_2,$$

under some composition operator $\|$. This problem has been approached for several types of behavioral relationships such as simulation relations in transition systems [TPL04, Fre05] and in Moore machines [HQRT98], approximate bisimulation [TI08] and alternating approximate simulation [RT]. In this chapter, we approach the problem from another point of view since we do not assume that we start from independent systems or a prior decomposition of a system. Instead, we start from a global system which is too large to allow the use of the symbolic abstraction and control methods described above and we provide a method to decompose it into subsystems of more reasonable dimensions. The symbolic methods are then applied to each subsystem and refined into a strategy for the global system. Thus, we do not only focus on proving that the simulation relation is preserved under composition, but we actually provide a systematic method for controller synthesis of a large scale system to realize safety specification associated with performance guarantees. In addition, we do not require any prior decomposable structure of the global system and the synthesized controller remains correct by construction for any choice of the subsystems. We should note however that a poorly chosen decomposition where strongly coupled states are split into two different subsystems may lead to an empty controller. Our method is based on two assumptions similar to an assume-guarantee reasoning: for each subsystem,

- unmodeled state components do not violate their safety specification;

- state components that are modeled but not controlled do not violate their safety specification.

To the best of our knowledge, there is almost no work on abstraction-based methods going in this direction of decomposing a large problem into simpler ones. We could only find [Rei10], where the focus is on the control of a particular class of systems which are decomposable into subsystems that share a common control input and whose states are not coupled.

4.2 Notations

Indices Even more than in the previous chapter, many alphanumerical indices are used on the variables, functions or sets to denote, for example, a component of a vector variable, a discrete time instant or some naming information. Until now, we used the following rules as much as possible.

- Numerical indices and letters i and j are used as *subscripts* to refer to a component of a variable or function, or the projection of a set on the corresponding dimension (e.g. let $X = [\underline{x}, \overline{x}] \subseteq \mathbb{R}^n$, then $X_i = [\underline{x}_i, \overline{x}_i] \subseteq \mathbb{R}$ is the scalar interval on the i^{th} dimension).
- Numerical indices and letters j and k are used as *superscripts* to refer to discrete time instants (e.g. x^k is the k^{th} element of a sequence $(x^0, x^1, x^2, \dots)$).
- Other alphabetical indices, the number 0 and symbols such as $*$ are simply used for naming and have no predefined position.

With the introduction in this chapter of new indices for the subsystems and sets of indices representing the states or inputs of interest in a subsystem, some modifications of the first rule are necessary. From now on, a set of indices used as a subscript corresponds to the extension of the first rule (if $I = \{1, 3, 4\}$ and $x \in \mathbb{R}^4$, then $x_I = (x_1, x_3, x_4)$). Alternatively, when the notations are too complicated for the use of a subscript to be sufficiently clear, we may use the projection operator π_I on a set of dimensions of the appropriate space (e.g. $\pi_I([\underline{x}, \overline{x}]) = [\underline{x}_I, \overline{x}_I]$). On the other hand, scalar indices now simply become a naming information relating a variable, function or set to the subsystem of same index (e.g. for $I \subseteq \mathbb{N}$ and $i \in \mathbb{N}$, $u_I = \pi_I(u)$ is a subset of the components of an input named u, while u_i is an input of subsystem S_i and is not related to u).

Decomposition Consider that we want to decompose our system into $m \in \mathbb{N}$ subsystems. We need to introduce six index sets describing the state and input components related to each subsystem. Let $(I_1^c, \dots, I_m^c)$ be a partition of the set of state indices $\{1, \dots, n\}$. For subsystem $i \in \{1, \dots, m\}$, we consider the following four index sets:

- $I_i \supseteq I_i^c$ represents all the state components whose dynamics are modeled in the subsystem;
- I_i^c are the state components to be controlled;

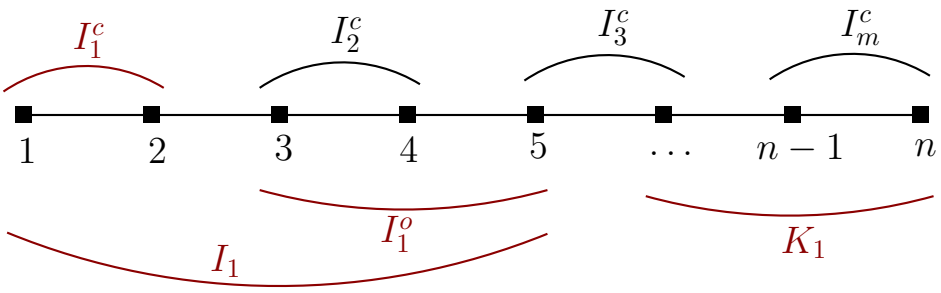


Figure 4.1 – Partition of $\{1,\dots,n\}$ and state index sets for subsystem 1.

- $I_i^o = I_i \backslash I_i^c$ are the state components that are only observed but not controlled;
- $K_i = \{1,\dots,n\}\backslash I_i$ are the remaining unobserved state components considered as external inputs.

These sets are illustrated in Figure 4.1 where we can see the initial partition of $\{1,\dots,n\}$ and the three other sets for subsystem 1. We only have two index sets for the control input as we consider that all control inputs of a subsystem are actually used for control and having an input common to two subsystems may lead to incompatible strategies. Let $(J_1,\dots,J_m)$ be a partition of the set of control input indices $\{1,\dots,p\}$. For subsystem $i \in \{1,\dots,m\}$, we consider the following index sets:

- J_i are the indices of the control inputs used to control the states $x_{I_i^c}$;
- $L_i = \{1,\dots,p\}\backslash J_i$ are the remaining control components considered as external inputs.

The role of all these index sets can be summarized as follows: for subsystem $i \in \{1,\dots,m\}$, we model the states $x_{I_i} = (x_{I_i^c}, x_{I_i^o})$ where $x_{I_i^c}$ are to be controlled using the inputs u_{J_i} and $x_{I_i^o}$ are simply observed to increase the precision of the subsystem while x_{K_i} and u_{L_i} are considered as external disturbances.

Composition Since $(I_1,\dots,I_m)$ is a covering of $\{1,\dots,n\}$, some state variables may be modeled in several subsystems. When composing these subsystems, we need to synchronize the values of the state components appearing more than once. For the composition of sets of states, we thus introduce a new operator $\Cap$ that is halfway between the Cartesian product and the classical set intersection. Consider two sets $X, Y \subseteq \mathbb{R}^n$, two index sets $I, J \subseteq \{1,\dots,n\}$ and the projections $X_I = \pi_I(X)$ and $Y_J = \pi_J(Y)$ of X and Y on lower dimensional spaces. The operator $\Cap$ is defined as follows:

$$X_I \Cap Y_J = \{z \in \pi_{I\cup J}(\mathbb{R}^n) \mid z_I \in X_I,\ z_J \in Y_J\}. \tag{4.1}$$

We can see that when $I \cap J = \emptyset$, X_I and Y_J have no dimension in common and this operator is equivalent to the Cartesian product: $\Cap \equiv \times$. On the other hand, when all the dimensions of X_I and Y_J are the same ($I = J$), then (4.1) gives the classical set intersection: $\Cap \equiv \cap$.

4.3 Subsystems

Unlike Chapter 3 where we do not need cooperativeness with respect to the control input (Assumption 1′), here we consider that Assumption 1 is satisfied throughout this chapter: the control and disturbance inputs are bounded and the continuous system (1.1) is cooperative with respect to the state, the control input and the disturbance input. Other assumptions are not required but we can note that such compositional approaches are particularly interesting for systems satisfying Assumption 2 where each control input only affects a single state variable: for any decomposition of the states, there will necessarily be no dependence on the control inputs of other subsystems.

4.3.1 Abstractions

The dynamics of the continuous system (1.1) are decomposed into $m \in \mathbb{N}$ partial representations of (1.1) where some of the state and input components are not observed. Each of these partial descriptions then is abstracted into a symbolic subsystem following the method in Section 3.3. As described in Section 4.2, in these subsystems some of the states or control inputs are considered as external disturbances (indices K_i and L_i, respectively). In Section 3.3, the creation of the symbolic abstraction S_a requires the disturbance input w to be bounded, provided by Assumption 1′. We thus need a similar assumption on x_{K_i} and u_{L_i} for subsystem i. Assumption 1 already provides $u_{L_i} \in \pi_{L_i}([\underline{u}, \overline{u}])$. For the unobserved state components x_{K_i}, we need to introduce a first assume-guarantee obligation allowing a subsystem to be modeled under the assumption that the safety specifications on the external state components are realized by other subsystems.

A/G Obligation 1. *For all* $i \in \{1, \dots, m\}$, $x_{K_i} \in \pi_{K_i}([\underline{x}, \overline{x}])$.

As described in Section 4.2, only a part of the state x_{I_i} modeled in the i^{th} subsystem is to be controlled ($x_{I_i^c}$) while the other components ($x_{I_i^o}$) are only observed to improve the precision of the model. For this reason, we need to introduce a second assume-guarantee obligation where we consider that for each subsystem, the safety specifications of the observed but uncontrolled states are realized.

A/G Obligation 2. *For all* $i \in \{1, \dots, m\}$, $x_{I_i^o} \in \pi_{I_i^o}([\underline{x}, \overline{x}])$.

We can then define subsystem i denoted as $S_i = (X_i, X_i^0, U_i, \underset{i}{\longrightarrow})$ and composed of the following elements:

- $X_i^0 = \mathcal{P}_{I_i}^0 = \pi_{I_i}(\mathcal{P}^0)$ is a uniform partition of $\pi_{I_i}([\underline{x}, \overline{x}])$ as in (3.2);
- $X_i = X_i^0 \cup \{Out_i\}$ is a partition of $\pi_{I_i}(\mathbb{R}^n)$, similarly to (3.3);
- $U_i = U_{J_i}^d = \pi_{J_i}(U^d)$ where U^d is the discretized control input set (3.4).

Before defining the transitions of S_i, we denote as $RS_i(s_i, u_i)$ the over-approximation of the reachable set of (1.1) in $\mathbb{R}^n$ from $s_i \in X_i^0$ with $u_i \in U_i$ and under Assumption 1 and A/G Obligation 1:

$$RS_i(s_i, u_i) = [\Phi(\tau, (\underline{s_i}, \underline{x}_{K_i}), (u_i, \underline{u}_{L_i}), \underline{w}), \Phi(\tau, (\overline{s_i}, \overline{x}_{K_i}), (u_i, \overline{u}_{L_i}), \overline{w})]. \tag{4.2}$$

Then, the transitions of S_i can be defined as follows:

- $\forall s_i \in X_i^0,\ u_i \in U_i,\ s_i' \in X_i^0,\ s_i \xrightarrow[i]{u_i} s_i' \iff s_i' \cap \pi_{I_i}(RS_i(s_i, u_i)) \neq \emptyset$;
- $\forall s_i \in X_i^0,\ u_i \in U_i,\ s_i \xrightarrow[i]{u_i} Out_i \iff \pi_{I_i^c}(RS_i(s_i, u_i)) \not\subseteq \pi_{I_i^c}([\underline{x}, \overline{x}))$ or $\pi_{I_i}(RS_i(s_i, u_i)) \cap \pi_{I_i}([\underline{x}, \overline{x})) = \emptyset$;
- $\forall u_i \in U_i,\ s_i' \in X_i,\ Out_i \xrightarrow[i]{u_i} s_i'$.

In the first transition definition ($s_i, s_i' \in X_i^0$), we can clearly see from the monotonicity property (Definition 1.5) that the obtained over-approximation of the reachable set (4.2) is larger than for S_a due to the fact that we now have to consider the worst-case values of the new disturbances x_{K_i} and u_{L_i}. The second point ($s_i \in X_i^0$, $s_i' = Out_i$) states that a transition to Out_i exists either if the reachable set (4.2) leaves the safety specification on the dimensions on the controlled states (indices I_i^c), or if the pair (s_i, u_i) has no other successor ($Post_i(s_i, u_i) = Out_i$). These conditions are obtained by combining a condition similar to the first point (with $s_i' = Out_i$) with A/G Obligation 2. This case is explained in details in Example 4.1 below. Finally, the third point of the transition definition ($s_i = Out_i$) is the same as in S_a where we take $Post_i(Out_i, u_i) = X_i$ for all $u_i \in U_i$ in order to ensure the alternating simulation proven in the next section.

We consider an example to help us pinpoint the conditions where A/G Obligation 2 has an effect on the definition of the transitions of S_i.

Example 4.1. Consider a subsystem S_i with $I_i = \{1, 2\}$, $I_i^c = 1$ and $I_i^o = 2$. The state space of S_i is represented in Figure 4.2 where $x_1 = x_{I_i^c}$ and $x_2 = x_{I_i^o}$ correspond to the horizontal and vertical axes, respectively. In this figure, we also give 9 possible positions of the over-approximation of the reachable set $RS_i(s_i, u_i)$ defined in (4.2) for some $s_i \in X_i^0$ and $u_i \in U_i$. There are 4 possible behaviors covering all 9 intervals in Figure 4.2.

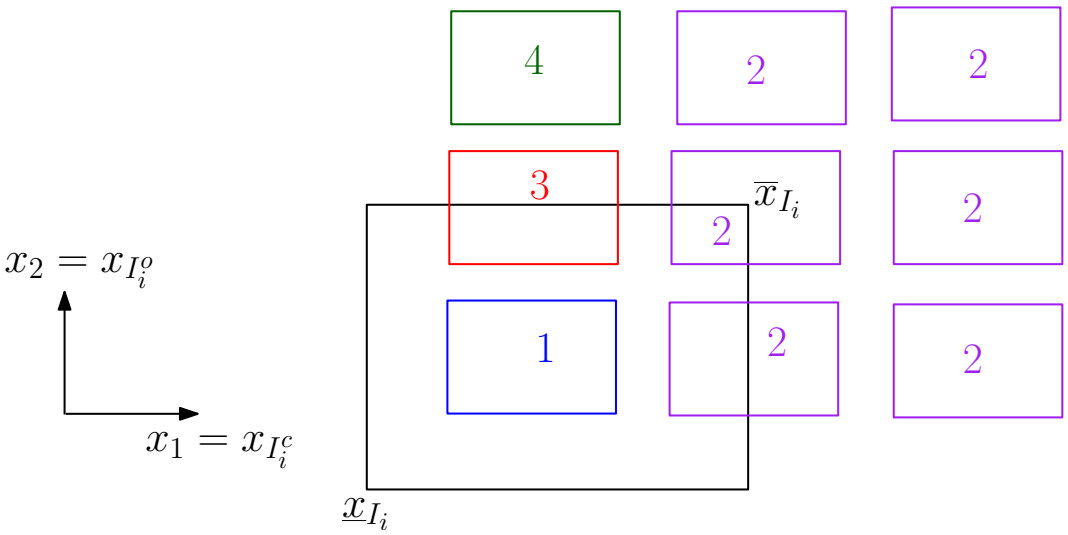


Figure 4.2 – Possible over-approximations of the reachable set used in S_i.

1. For the bottom left case (blue interval), $RS_i(s_i, u_i)$ is included in $[\underline{x}_{I_i}, \overline{x}_{I_i})$ which means that A/G Obligation 2 is not needed and the transitions are defined as in Chapter 3 (with the new reachable set $RS_i(s_i, u_i)$).

2. For all six cases on the right side of Figure 4.2 (purple intervals), $RS_i(s_i, u_i)$ violates the safety specification on the dimension I_i^c of the controllable state. Hence, even with A/G Obligation 2, the transition $s_i \xrightarrow[i]{u_i} Out_i$ exists in S_i.

3. In the middle left case (red interval), $RS_i(s_i, u_i)$ only violates the specifications on the dimension I_i^o of the uncontrolled state. With A/G Obligation 2, we thus know that $Out_i \notin Post_i(s_i, u_i)$ which means that $Post_i(s_i, u_i) \subseteq X_i^0$.

4. Lastly, for the top left case (green interval), we can see that A/G Obligation 2 is not compatible with the reachable set $RS_i(s_i, u_i)$ which lands completely outside of the target interval. We thus know that the pair (s_i, u_i) is unsafe and we keep Out_i as the only successor: $Post_i(s_i, u_i) = \{Out_i\}$.

Compared to the classical method in Chapter 3 or in the first point of the transition definition for S_i ($s_i, s_i' \in X_i^0$), we can notice that A/G Obligation 2 only has an influence in case 3, where the obligation prevents a transition to Out_i. Case 3 corresponds to the condition where $RS_i(s_i, u_i)$ only violates the safety specification on the uncontrolled state dimension I_i^o and Out_i is not the only successor. The second point of the transition definition for S_i is the negation of that statement: $Out_i \in Post_i(s_i, u_i)$ if and only if $RS_i(s_i, u_i)$ violates the safety specification on the controlled state dimension I_i^c or if Out_i is the only successor. Note that with this definition, the set $Post_i(s_i, u_i)$ is never empty and we can use $U_i(s_i) = U_i$ as in Remark 3.2. △

4.3.2 Controller synthesis

Safety Solving the safety game that keeps the symbols of S_i in $X_i^0 = \pi_{I_i}(\mathcal{P}^0)$ is achieved using the method presented in Section 3.4.1. Note that for a subsystem i, the real control objective only is to realize the safety specifications for the controlled state components (indices I_i^c), but with A/G Obligation 2 used to define S_i this is equivalent to solving the safety game for all modeled states (indices I_i). Similarly to (3.6), we define the operator $F_{X_i^0} : 2^{X_i} \to 2^{X_i}$ corresponding to the safety game in X_i^0 applied on subsystem S_i:

$$F_{X_i^0}(Z) = \{s_i \in Z \cap X_i^0 \mid \exists u_i \in U_i,\ Post_i(s_i, u_i) \subseteq Z\}. \tag{4.3}$$

The maximal fixed-point of $F_{X_i^0}$ is denoted as Z_i and can also be obtained in a finite number of steps since S_i is a finite transition system. We can then define the associated safety controller $C_i : Z_i \to 2^{U_i}$ ensuring that S_i stays at all time in the safe set Z_i:

$$C_i(s_i) = \{u_i \in U_i \mid \emptyset \neq Post_i(s_i, u_i) \subseteq Z_i\}. \tag{4.4}$$

Receding horizon Let $g_i : Z_i \times U_i \to \mathbb{R}^+$ be a cost function of S_i such that

$$g_i(s_i, u_i) = g_i(s'_i, u_i) \text{ if } \pi_{I_i^c}(s_i) = \pi_{I_i^c}(s'_i). \tag{4.5}$$

In (4.5), the cost function g_i is chosen to be independent of the uncontrolled state components (indices $I_i^o = I_i \backslash I_i^c$). This comes from the same reason that lead to A/G Obligation 2: the control objectives on S_i only concern the behavior in $\pi_{I_i^c}(X_i)$ and it is natural that the measure of the performances is not affected by the uncontrolled components.

As in Section 3.4.2, for a trajectory $(s_i^0, u_i^0, s_i^1, \ldots, s_i^N)$ of S_i controlled with C_i over a finite time horizon of size N, we want to minimize the cost

$$\sum_{k=0}^{N} \lambda^k g_i(s_i^k, u_i^k). \tag{4.6}$$

This is done with a dynamic programming algorithm formulated as follows:

$$J_i^N(s_i) = \min_{u_i \in C_i(s_i)} g_i(s_i, u_i), \tag{4.7a}$$

$$J_i^k(s_i) = \min_{u_i \in C_i(s_i)} \left(g_i(s_i, u_i) + \lambda \max_{s'_i \in Post_i(s_i, u_i)} J_i^{k+1}(s'_i) \right). \tag{4.7b}$$

Then we use a receding horizon control scheme on the control policy provided by (4.7) to obtain a deterministic controller $C_i^* : Z_i \to U_i$ for S_i:

$$C_i^*(s_i) = \arg\min_{u_i \in C_i(s_i)} \left(g_i(s_i, u_i) + \lambda \max_{s'_i \in Post_i(s_i, u_i)} J_i^1(s'_i) \right). \tag{4.8}$$

4.4 Composition

Now that for each subsystem we have obtained a safe set Z_i, the associated safety controller $C_i : Z_i \to 2^{U_i}$ and the deterministic controller $C_i^* : Z_i \to U_i$ minimizing (4.6) with worst-case predictions of future steps, we need to make sure that the composition of these controllers realizes the safety specification for the original system S and that the optimization provides some performance guarantees as in Theorem 3.10. Let the transition system $S_c = (X_c, X_c^0, U_c, \underset{c}{\longrightarrow})$ be the result of the composition of the subsystems S_i for all $i \in \{1, \ldots, m\}$. S_c contains the following elements:

- $X_c^0 = X_1^0 \Cap \cdots \Cap X_m^0 = \mathcal{P}^0$;
- $X_c = X_c^0 \cup \{Out\} = \mathcal{P}$;
- $U_c = U_1 \times \cdots \times U_m = U^d$;
- $\forall s \in \mathcal{P}^0,\ u \in U^d,\ s' \in \mathcal{P}^0,\ s \xrightarrow[c]{u} s' \iff \forall i \in \{1, \ldots, m\},\ s_{I_i} \xrightarrow[i]{u_{J_i}} s'_{I_i}$
- $\forall s \in \mathcal{P}^0,\ u \in U^d,\ s \xrightarrow[c]{u} Out \iff \exists i \in \{1, \ldots, m\} \mid s_{I_i} \xrightarrow[i]{u_{J_i}} Out_i$

- $\forall u \in U^d,\ s' \in \mathcal{P},\ Out \xrightarrow[c]{u} s'$.

The set of initial states is obtained by composing the sets X_i^0 with the operator defined in (4.1). The set of states X_c is not taken as the composition of the sets X_i since we do not need more than one symbol to represent the exterior of the target interval $[\underline{x}, \overline{x})$. For U_c, the composition can simply use the Cartesian product since the index sets $(J_1, \ldots, J_m)$ form a partition of $\{1, \ldots, p\}$. For a safe transition in S_c ($s, s' \in \mathcal{P}^0$), we need the transition to exist in all subsystems S_i using the projections of s, s' and u on the appropriate dimensions. On the other hand, to have a transition of S_c going to the unsafe symbol Out, it suffices that one subsystem S_i has an unsafe transition. Finally, as in the definition of the other symbolic models S_a and S_i, we consider that all transitions from the symbol Out exist in S_c. We can then prove that this system is alternatingly simulated by the original system S.

Proposition 4.1. *Under Assumption 1, the identity function on $\mathcal{P}$ is an alternating simulation relation from S_c to S_a. Therefore, we have $S_c \preceq_{\mathcal{AS}} S$.*

Proof. For $S_c \preceq_{\mathcal{AS}} S_a$, the first condition is immediately satisfied since $X_c^0 = X_a^0$. For the second condition, let $s \in X_c = X_a = \mathcal{P}$, $u \in U_c(s) \subseteq U_a$ and $s' \in Post_a(s, u)$. If $s = Out$, we have $s' \in Post_c(Out, u) = \mathcal{P}$. If $s \in \mathcal{P}^0$, the transition in S_a is defined by $s' \cap [\Phi(\tau, \underline{s}, u, \underline{w}), \Phi(\tau, \overline{s}, u, \overline{w})] \neq \emptyset$. With the cooperativeness of (1.1) from Assumption 1, Definition 1.5 gives for all $i \in \{1, \ldots, m\}$:

$$\begin{cases} \Phi(\tau, (\underline{s}_{I_i}, \underline{x}_{K_i}), (u_{J_i}, \underline{u}_{L_i}), \underline{w}) \leq \Phi(\tau, \underline{s}, u, \underline{w}), \\ \Phi(\tau, (\overline{s}_{I_i}, \overline{x}_{K_i}), (u_{J_i}, \overline{u}_{L_i}), \overline{w}) \geq \Phi(\tau, \overline{s}, u, \overline{w}). \end{cases} \tag{4.9}$$

The transition $s' \in Post_a(s, u)$ thus implies for all $i \in \{1, \ldots, m\}$:

$$s' \cap [\Phi(\tau, (\underline{s}_{I_i}, \underline{x}_{K_i}), (u_{J_i}, \underline{u}_{L_i}), \underline{w}), \Phi(\tau, (\overline{s}_{I_i}, \overline{x}_{K_i}), (u_{J_i}, \overline{u}_{L_i}), \overline{w})] \neq \emptyset. \tag{4.10}$$

If $s' \in \mathcal{P}^0$, then for all $i \in \{1, \ldots, m\}$, (4.10) gives $s'_{I_i} \in Post_i(s_{I_i}, u_{J_i})$ with $s_{I_i}, s'_{I_i} \in X_i^0$. Then we obtain $s' \in Post_c(s, u)$ from the definition of S_c.

If $s' = Out$, the transition $Out \in Post_a(s, u)$ means that there exists a dimension $j \in \{1, \ldots, n\}$ such that $\pi_j([\Phi(\tau, \underline{s}, u, \underline{w}), \Phi(\tau, \overline{s}, u, \overline{w})]) \not\subseteq \pi_j([\underline{x}, \overline{x}))$. With the over-approximation (4.9), it gives for all $i \in \{1, \ldots, m\}$:

$$\pi_j([\Phi(\tau, (\underline{s}_{I_i}, \underline{x}_{K_i}), (u_{J_i}, \underline{u}_{L_i}), \underline{w}), \Phi(\tau, (\overline{s}_{I_i}, \overline{x}_{K_i}), (u_{J_i}, \overline{u}_{L_i}), \overline{w})]) \not\subseteq \pi_j([\underline{x}, \overline{x})).$$

This means that for the subsystem i such that $j \in I_i^c$, we have the transition $Out_i \in Post_i(s_{I_i}, u_{J_i})$ (case 2 of Figure 4.2 and Example 4.1), which implies that $s' = Out \in Post_c(s, u)$. Thus we have $S_c \preceq_{\mathcal{AS}} S_a$ with the identity function H_c on $\mathcal{P}$. For the second part of the result, we use the fact that $S_a \preceq_{\mathcal{AS}} S$ from Proposition 3.6 and the transitivity of the alternating simulation proven in Proposition 3.5. The alternating simulation relation from S_c to S is the composition $H_c \circ H_a = H_a$. □

Using some parts of the proof of Proposition 4.1, we can also show that S_c satisfies Remark 3.2.

Corollary 4.2. *For all $s \in \mathcal{P}$, $u \in U^d$, $Post_c(s, u) \neq \emptyset$ and $U_c(s) = U^d$.*

Proof. If $s = Out$, we immediately have $Post_c(Out, u) = \mathcal{P} \neq \emptyset$. Let $s \in \mathcal{P}^0$ and $u \in U^d$. Since all S_i satisfy Remark 3.2, $Post_i(s_{I_i}, u_{J_i}) \neq \emptyset$ for all $i \in \{1, \ldots, m\}$. If there exists a subsystem S_i such that $Out_i \in Post_i(s_{I_i}, u_{J_i})$, then by definition of S_c we have $Out \in Post_c(s, u) \neq \emptyset$. Otherwise, we have $Post_i(s_{I_i}, u_{J_i}) \subseteq X_i^0$ for all i and using the over-approximation (4.9), this means that $Post_a(s, u) \subseteq \mathcal{P}^0$. Since S_a also satisfies Remark 3.2, we have $Post_a(s, u) \neq \emptyset$. Then in these conditions, $s' \in Post_a(s, u)$ and (4.10) implies that $s'_{I_i} \in Post_i(s_{I_i}, u_{J_i})$ for all i, which gives $s' \in Post_c(s, u)$. □

4.4.1 Safety

Similarly to X_c^0 and U_c, we can compose the safe sets Z_i and safety controllers C_i of all subsystems using the operator $\Cap$ for the state sets and the Cartesian product for the input sets:

$$Z_c = Z_1 \Cap \cdots \Cap Z_m, \tag{4.11}$$

$$\forall s \in Z_c,\ C_c(s) = C_1(s_{I_1}) \times \cdots \times C_m(s_{I_m}). \tag{4.12}$$

To use them with the original system S, we can give their equivalent form Z_c^X and $C_c^X : Z_c^X \to 2^U$ after a projection to the continuous state space $X = \mathbb{R}^n$:

$$Z_c^X = \{x \in \mathbb{R}^n \mid \exists s \in Z_c,\ x \in s\}, \tag{4.13}$$

$$\forall x \in Z_c^X,\ C_c^X(x) = C_c(H_a(x)), \tag{4.14}$$

where $H_a : \mathbb{R}^n \to \mathcal{P}$ is the alternating simulation relation defined in Proposition 3.6. We can then prove that Z_c^X is a safe set of S and C_c^X is a safety controller for S, thus solving Control Problem 1.

Theorem 4.3. *$Z_c^X \subseteq [\underline{x}, \overline{x})$ is a safe set for system S controlled with any strategy of C_c^X.*

Proof. Let $s \in Z_c$, $u \in C_c(s)$ and $s' \in Post_c(s, u)$. By construction of C_i (4.4), we have $Post_i(s_{I_i}, u_{J_i}) \subseteq Z_i$ for all $i \in \{1, \ldots, m\}$, which implies that $s' \in Z_1 \Cap \cdots \Cap Z_m = Z_c$. Then, Z_c is a safe set for S_c controlled with C_c. For $x \in Z_c^X$, $s = H_a(x) \in Z_c$, $u \in C_c^X(x) = C_c(s)$ and $x' \in Post(x, u)$, the alternating simulation from Proposition 4.1 implies that $H_a(x') \in Post_c(s, u) \subseteq Z_c$. Therefore, $x' \in Z_c^X$ and C_c^X is a safety controller for S in Z_c^X. □

If instead of S we look at controlling S_a with C_c, we can also obtain the following result.

Corollary 4.4. *$Z_c \subseteq Z_a$.*

Proof. If in the proof of Theorem 4.3 we use the alternating simulation $S_c \preceq_{\mathcal{AS}} S_a$ instead of $S_c \preceq_{\mathcal{AS}} S$, we immediately show that Z_c is a safe set for the safety game on S_a. Then, the result is obtained from the fact that Z_a is the maximal safe set of S_a. □

In the following example, we illustrate this corollary for two choices of decomposition.

Example 4.2. Consider the coupled-tank system introduced in Section 1.3.2. In Example 3.2 on the centralized symbolic method (Section 3.4.1), we worked with the safety specification $x_2 \in [15, 20)$. Since the compositional approach is more conservative due to the lack of information in each subsystem, the safety specification are not realizable in this target interval which is only partially safe for the centralized method. We thus need to relax our specifications: we know want $x_2 \in [10, 25)$. In the subsystem centered on tank 2, no information on the level x_1 or on inflow u_1 in tank 1 are available. To ensure a minimal inflow from tank 1, we thus add the specifications $x_1 \in [10, 30)$. The symbolic abstractions are created with $\alpha_x = 10$, $\alpha_u = 4$ and $\tau = 0.5\,\text{s}$.

In these conditions, the centralized method from Section 3.4.1 gives a safe set containing all the symbols: $Z_a = \mathcal{P}^0$ and $Z_a^X = [\underline{x}, \overline{x})$. Then, for the compositional approach, we consider two possible decompositions. In the first one, each subsystem focuses on one tank and abstracts everything else: $I_i = I_i^c = J_i = i$ and $I_i^o = \emptyset$ for all $i \in \{1, 2\}$. The second one is similar apart from the fact that we take 2-dimensional subsystems where the state of the other tank is modeled but not controlled: $I_i^c = J_i = i$ and $I_i = \{1, 2\}$ for all $i \in \{1, 2\}$. In both cases, subsystems 1 is fully safe: keeping the water level x_1 in $[10, 30)$ is independent of what happens in the tank 2. On the other hand, the water level in tank 2 is more easily controlled when we know the current level in tank 1 and the value of the control of the first pump. In the compositional method using $2D$ subsystems, the safety specification in S_2^{2D} can be realized if the water level in tank 1 is not too high ($x_1 \leq 24\,\text{cm}$) since it would create too much inflow in tank 2 to stay below its upper bound. However, if we use $1D$ subsystems, S_2^{1D} does not have any information on the actual value of x_1 and therefore always assumes the worst-case, leading to violating the safety specifications. As partially represented in Figure 4.3, we thus have $\emptyset = Z_c^{1D} \subsetneq Z_c^{2D} \subsetneq Z_a = \mathcal{P}^0$, which is what was expected from Corollary 4.4. △

Therefore, Corollary 4.4 and Example 4.2 indicate that the less information are taken to model the system or subsystems, the smaller the safe set realizing the safety specification. A similar result is obtained on the performance guarantees in Section 4.5.

4.4.2 Performance guarantee

The deterministic controllers $C_i^* : Z_i \to U_i$ can also be composed into $C_c^* : Z_c \to U_c$ and refined into a controller $C_c^{*X} : Z_c^X \to U$ of S using the alternating simulation from Proposition 4.1:

$$\forall s \in Z_c,\ C_c^*(s) = (C_1^*(s_{I_1}), \dots, C_m^*(s_{I_m})), \tag{4.15}$$

$$\forall x \in Z_c^X,\ C_c^{*X}(x) = C_c^*(H_a(x)). \tag{4.16}$$

Let M_i denote the maximal value of J_i^N in (4.7a) for subsystem S_i:

$$M_i = \max_{s_i \in Z_i} J_i^N(s_i) = \max_{s_i \in Z_i} \min_{u_i \in C_i(s_i)} g_i(s_i, u_i). \tag{4.17}$$

For the next result, we take the following assumption that compares the cost functions of S_a and S_i and the upper bounds M_a and M_i.

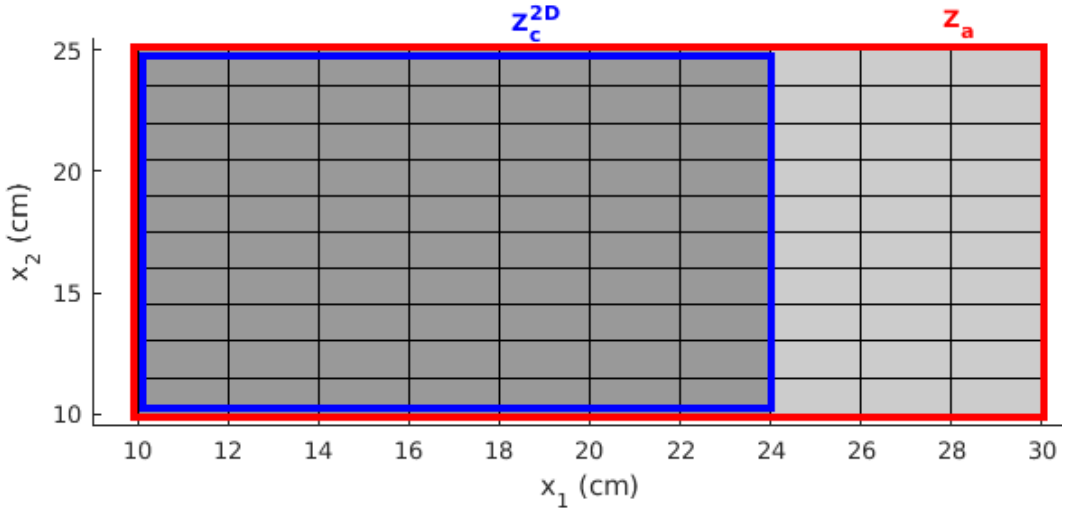


Figure 4.3 – Safe set Z_c^{2D} for the compositional approach with $2D$ subsystems (dark gray symbols in the blue rectangle) and safe set Z_a for the centralized approach (all symbols in the red rectangle).

Assumption 6. $\forall s \in Z_c,\ u \in U^d,\ g_a(s,u) \leq \sum_{i=1}^{m} g_i(s_{I_i}, u_{J_i}).\ M_a \leq \sum_{i=1}^{m} M_i.$

If the second part of Assumption 6 is not satisfied from the choice of the cost functions g_a and g_i, we can set the constants M_i with greater values than (4.17) to enforce Assumption 6. In that case, all the following results remain valid and we simply obtain less tight performance guarantees. Similarly to Theorem 3.10, we can then solve Control Problem 2 by providing an upper bound on the performance criterion (3.1) for any trajectory of S controlled with C_c^{*X}.

Theorem 4.5. *Let $(x^0, u^0, x^1, u^1, \dots)$ with $x^0 \in Z_c^X$ be a trajectory of S controlled with C_c^{*X} in (4.16). For all $k \in \mathbb{N}$, let $s^k = H_a(x^k)$. Then under Assumption 6, for all $k \in \mathbb{N}$ we have,*

$$\sum_{j=0}^{+\infty} \lambda^j g(x^{k+j}, u^{k+j}) \leq \sum_{i=1}^{m} J_i^0(s_{I_i}^k) + \frac{\lambda^{N+1}}{1-\lambda} \sum_{i=1}^{m} M_i.$$

Proof. Since for all $x \in Z_c^X$ we have $C_c^{*X}(x) \in C_c^X(x)$, then C_c^{*X} also is a safety controller and $x^k \in Z_c^X$ for all $k \geq 1$ if $x^0 \in Z_c^X$. Similarly to Lemma 3.9, we can show that for all $i \in \{1, \dots, m\}$ and $s_i \in Z_i$, we have $J_i^0(s_i) \leq J_i^1(s_i) + \lambda^N M_i$ where M_i is defined in (4.17). Then, as in the proof of Theorem 3.10, we can use this result in the definition of $J_i^0(s_i^k)$ (4.7b) with $u_i^k = C_i^*(s_i^k)$ to obtain

$$\sum_{j=0}^{+\infty} \lambda^j g_i(s_i^{k+j}, u_i^{k+j}) \leq J_i^0(s_i^k) + \frac{\lambda^{N+1}}{1-\lambda} M_i. \tag{4.18}$$

If we combine the definition of g_a (3.10) and Assumption 6, we have

$$\forall s \in Z_c,\ x \in s,\ u \in U^d,\ g(x,u) \leq g_a(s,u) \leq \sum_{i=1}^{m} g_i(s_{I_i}, u_{J_i}).$$

The inequality of Theorem 4.5 is then obtained by taking the sum of (4.18) over all subsystems S_i. □

4.5 Performance comparison

We have already shown in Corollary 4.4 that the safety controller synthesis with the centralized method in Section 3.4.1 yields better results than the compositional approach: $Z_c \subseteq Z_a$. In this section, we want to compare the performance guarantees provided in Theorems 3.10 and 4.5. For this, we need to introduce some new notations and intermediate results. Let the cost function $g_c : Z_c \times U_c \to \mathbb{R}^+$ be defined as the sum of the cost functions g_i from the subsystems:

$$g_c(s,u) = \sum_{i=1}^{m} g_i(s_{I_i}, u_{J_i}). \tag{4.19}$$

Note that since the function g_i only depends on the input and the controlled state components (indices I_i^c), each state and input component influences g_c exactly once. Consider the functions $J_c^k : Z_c \to \mathbb{R}^+$ for $k \in \{0,\dots,N\}$ used to solve the dynamic programming algorithm on S_c with the safety controller C_c and the cost functions in (4.19):

$$J_c^N(s) = \min_{u \in C_c(s)} g_c(s,u), \tag{4.20a}$$

$$J_c^k(s) = \min_{u \in C_c(s)} \left(g_c(s,u) + \lambda \max_{s' \in Post_c(s,u)} J_c^{k+1}(s') \right). \tag{4.20b}$$

Although the functions J_c^k will never actually be computed as it would defeat the purpose of the compositional approach, (4.20) provides useful notations for the next results. From the definition of the dynamic programming on S_i and S_c, we show that the function J_c^k is smaller than the sum of the functions J_i^k in (4.7).

Proposition 4.6. $\forall s \in Z_c,\ k \in \{0,\dots,N\},\ J_c^k(s) \leq \sum_{i=1}^{m} J_i^k(s_{I_i})$.

Proof. This is proven by induction. Since $(J_1,\dots,J_m)$ is a partition of $\{1,\dots,m\}$ and g_i defined in (4.5) only depends on the controlled components of the symbol, we have

$$J_c^N(s) = \min_{u \in C_c(s)} g_c(s,u) = \sum_{i=1}^{m} \min_{u_{J_i} \in C_i(s_{I_i})} g_i(s_{I_i}, u_{J_i}) = \sum_{i=1}^{m} J_i^N(s_{I_i}).$$

Now assume that we have $J_c^{k+1}(s) \leq \sum_{i=1}^{m} J_i^{k+1}(s_{I_i})$ for some $k \in \{0,\dots,N-1\}$. Using this assumption and the definition of g_c (4.19) in the next step of the dynamic programming algorithm on S_c (4.20b), we obtain:

$$J_c^k(s) \leq \min_{u \in C_c(s)} \left(\sum_{i=1}^{m} g_i(s_{I_i}, u_{J_i}) + \lambda \max_{s' \in Post_c(s,u)} \sum_{i=1}^{m} J_i^{k+1}(s'_{I_i}) \right). \tag{4.21}$$

In the composition of the subsystems to obtain a transition $s \xrightarrow[c]{u} s'$ in S_c, we require the existence in each subsystem S_i of the transition $s_{I_i} \xrightarrow[i]{u_{J_i}} s'_{I_i}$. If for $i \neq j$ we have $I_i \cap I_j \neq \emptyset$, this means that we have to synchronize the successors of subsystems S_i and S_j: $\pi_{I_i \cap I_j}(s_{I_i}) = \pi_{I_i \cap I_j}(s_{I_j})$. As a consequence, taking the maximum over the successors $s' \in Post_c(s,u)$ of S_c is necessarily more restrictive than taking the maxima over the successors $s'_{I_i} \in Post_i(s_{I_i}, u_{J_i})$ of the subsystems separately. Then we have:

$$\max_{s' \in Post_c(s,u)} \sum_{i=1}^{m} J_i^{k+1}(s'_{I_i}) \leq \sum_{i=1}^{m} \max_{s'_{I_i} \in Post_i(s_{I_i}, u_{J_i})} J_i^{k+1}(s'_{I_i}), \tag{4.22}$$

which gives in (4.21):

$$J_c^k(s) \leq \min_{u \in C_c(s)} \sum_{i=1}^{m} \left(g_i(s_{I_i}, u_{J_i}) + \lambda \max_{s'_{I_i} \in Post_i(s_{I_i}, u_{J_i})} J_i^{k+1}(s'_{I_i}) \right).$$

The condition $u \in C_c(s)$ can be decomposed into the m independent conditions $u_{J_i} \in C_i(s_{I_i})$. Since for $i \in \{1, \ldots, m\}$, the interior of the sum is independent of u_{J_j} for all $j \neq i$, we can switch the *min* and the $\sum$ operators to obtain the expected result using the definition of J_i^k in (4.7b). □

Remark 4.7. *When the subsystems are decoupled in the state ($I_i^o = \emptyset$ or equivalently $I_i = I_i^c$), the composition of their transitions does not require any synchronization. This leads to an equality in (4.22) as well as in the result of Proposition 4.6: $J_c^k(s) = \sum_{i=1}^{m} J_i^k(s_{I_i})$.*

Remark 4.7 is not true in the general case ($I_i^o \neq \emptyset$) as shown in the following counter-example.

Example 4.3. Consider a system with two state components and a partition of the target interval into four symbols as in Figure 4.4 where dimensions 1 and 2 are the horizontal and vertical axes, respectively. Let $I_1^c = 1$, $I_2^c = 2$ and $I_1 = I_2 = \{1, 2\}$. The partition of the control input indices is not detailed as it does not intervene in this example. Assume that for some $s \in \mathcal{P}^0$ and $u \in U^d$ we have $Post_1(s, u_{J_1}) = \{s_{11}, s_{12}, s_{21}, s_{22}\}$ and $Post_2(s, u_{J_1}) = \{s_{11}, s_{12}\}$. An illustration of a situation possibly leading to these successors is given in Figure 4.4 where $Post_a$, $Post_1$ and $Post_2$ are shortcut notations for the over-approximations of the reachable set from symbol s and with input u of systems S_a, S_1 and S_2 respectively.

The composition of the transitions in $Post_1(s, u_{J_1})$ and $Post_2(s, u_{J_2})$ requires the synchronization of the successors since both subsystems are defined on all the state dimensions ($I_1 = I_2 = \{1, 2\}$). We thus have $Post_c(s, u) = \{s_{11}, s_{12}\}$. Let the cost functions $g_1(s_{ij}, u_{J_1}) = i$ and $g_2(s_{ij}, u_{J_2}) = j$ which satisfy the requirement in (4.5) that g_i only depends on the symbol components of indices in I_i^c. With $g_c(s_{ij}, u) = g_1(s_{ij}, u_{J_1}) + g_2(s_{ij}, u_{J_2}) = i + j$ from (4.19), we have

$$\max_{s' \in Post_1(s, u_{J_1})} J_1^N(s') = \max_{s' \in Post_2(s, u_{J_2})} J_2^N(s') = 2 \; ; \; \max_{s' \in Post_c(s,u)} J_c^N(s') = 1 + 2,$$

which gives a strict inequality in (4.22) for $k = N - 1$. △

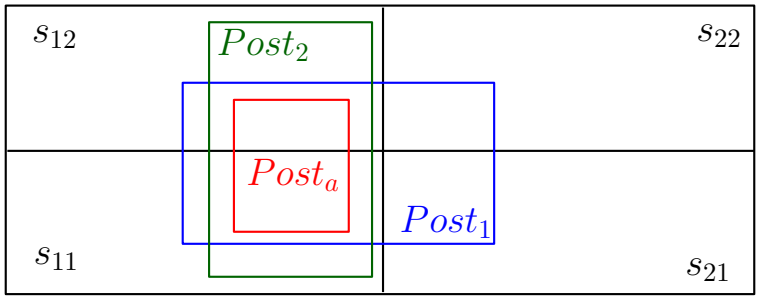


Figure 4.4 – State partition and illustration of the over-approximations of the reachable set for S_a, S_1 and S_2.

Proposition 4.6 can then be used to provide a comparison of the final cost J_a^0 obtained in (3.12) with the sum of the costs J_i^0 from (4.7).

Proposition 4.8. *Under Assumption 6, for all* $s \in Z_c$, $J_a^0(s) \leq \sum_{i=1}^{m} J_i^0(s_{I_i})$.

Proof. We first prove by induction that $J_a^k(s) \leq J_c^k(s)$ for all k. For the initial inequality, we consider the input $u \in U^d$ such that $J_i^N(s_{I_i}) = g_i(s_{i_i}, u_{J_i})$ for all $i \in \{1, \ldots, m\}$ in (4.7a). Then (3.12a) and the first part of Assumption 6 imply $J_a^N(s) \leq g_a(s,u) \leq \sum_{i=1}^{m} g_i(s_{i_i}, u_{J_i}) = J_c^N(s)$. Next, for all $k \in \{0, \ldots, N-1\}$, let G_a^k and u_a^k be defined as follows:

$$G_a^k(s,u) = g_a(s,u) + \lambda \max_{s' \in Post_a(s,u)} J_a^{k+1}(s'), \tag{4.23}$$

$$u_a^k = \arg\min_{u \in C_a(s)} G_a^k(s,u), \tag{4.24}$$

such that we have $J_a^k(s) = \min_{u \in C_a(s)} G_a^k(s,u) = G_a^k(s, u_a^k)$. We consider that there are similar notations G_c^k and u_c^k for S_c. Assume that for some $k \in \{0, \ldots, N-1\}$, $J_a^{k+1}(s) \leq J_c^{k+1}(s)$ for all $s \in Z_c$. Then we have:

$$\begin{aligned} J_a^k(s) = G_a^k(s, u_a^k) &\leq G_a^k(s, u_c^k) \\ &\leq g_c(s, u_c^k) + \lambda \max_{s' \in Post_a(s, u_c^k)} J_a^{k+1}(s') \\ &\leq g_c(s, u_c^k) + \lambda \max_{s' \in Post_c(s, u_c^k)} J_a^{k+1}(s') \\ &\leq g_c(s, u_c^k) + \lambda \max_{s' \in Post_c(s, u_c^k)} J_c^{k+1}(s') = G_c^k(s, u_c^k) = J_c^k(s). \end{aligned}$$

The first inequality comes from the definition of u_a^k, the second from the definition of g_c (4.19) and Assumption 6 ($g_a(s,u) \leq g_c(s,u)$), the third from the alternating simulation in Proposition 4.1 ($Post_a(s,u) \subseteq Post_c(s,u)$) and the last from the induction hypothesis. Finally, we combine this result with Proposition 4.6 and take $k = 0$. □

With the inequality in Proposition 4.8, we can then obtain a comparison of the performance guarantees provided in Theorems 3.10 and 4.5 by the centralized and compositional methods, respectively.

Corollary 4.9. *Under Assumption 6, we have for all $s \in Z_c$:*

$$J_a^0(s) + \frac{\lambda^{N+1}}{1-\lambda} M_a \leq \sum_{i=1}^{m} J_i^0(s_{I_i}) + \frac{\lambda^{N+1}}{1-\lambda} \sum_{i=1}^{m} M_i.$$

Proof. This is an immediate consequence of Proposition 4.8 and the second part of Assumption 6 ($M_a \leq \sum_{i=1}^{m} M_i$). □

Note that Corollary 4.9 only states that the performance guarantee obtained in Theorem 3.10 by using C_a^{*X} on S is better than when using C_c^{*X} (Theorem 4.5). This means that the actual value of the performance criterion (3.1) on a trajectory is upper bounded by a smaller value with the centralized method than with the compositional approach. This result thus does not provide a comparison of the actual performances and we can actually often observe in applications that in some conditions of the disturbances, the compositional approach gives smaller costs. In the following example, we illustrate the theorems on performance guarantees for both the centralized and compositional approaches and compare these guarantees as in Corollary 4.9.

Example 4.4. We consider the coupled-tank system from Section 1.3.2 in the same conditions than in Example 4.2: $\alpha_x = 10$, $\alpha_u = 4$, $\tau = 0.5\,\mathrm{s}$ and the safety specifications $x_1 \in [10, 30)$ and $x_2 \in [10, 25)$. We want to compare the performances and performance guarantees obtained with the centralized approach and the compositional approach using 2-dimensional subsystems. The approach with $1D$ subsystems from Example 4.2 is discarded as it does not realize the safety specification in these conditions.

As introduced in Section 1.3.2, the performance objective is to limit the use of both pumps, but with a bigger penalization for pump 2 which should only be used as a back-up. The associated cost function for the sampled system thus is $g(x, u) = u_1 + 5u_2$. Since g only depends on the control input, the associated cost function g_a is immediately obtained from (3.10):

$$g_a(s, u) = \max_{x \in s} g(x, u) = u_1 + 5u_2.$$

For the compositional approach, we take $g_1(s_1, u_1) = u_1$ and $g_2(s_2, u_2) = 5u_2$. We compute the values of M_a and M_i defined in (3.15) and (4.17), respectively: $M_a = 58.7$, $M_1 = 14.7$ and $M_2 = 110$. These functions satisfy Assumption 6, with an equality for the first part: $g_a(s, u) = g_1(s_1, u_1) + g_2(s_2, u_2)$.

For both methods, a deterministic controller is synthesized after using a dynamic programming algorithm over a finite window of $N = 5$ sampling periods and with a discount factor $\lambda = 0.5$. These values are chosen such that the first step that is neglected in the performance criterion on an infinite trajectory can be neglected: it has a factor $\lambda^{N+1} \approx 1.6\%$. Let J_a^0, J_1^0 and J_2^0 be the resulting functions of the dynamic programming applied to the centralized abstraction S_a and both subsystems S_1 and S_2. The minimum, maximum and average values of these functions and their difference are given in the first part of Table 4.1 for $s \in Z_c$. From the last line of this table, we can see that Proposition 4.8 is satisfied since the minimal value of the difference is positive. The average value of these functions naturally is higher for

$s \in Z_c$	$\min_{s\in Z_c}$	$\text{mean}_{s\in Z_c}$	$\max_{s\in Z_c}$
$J_a^0(s)$	19.94	39.62	107.48
$J_1^0(s_1) + J_2^0(s_2)$	47.44	111.08	245.44
$J_1^0(s_1) + J_2^0(s_2) - J_a^0(s)$	27.5	71.46	158.13
$J_a^0(s) + \frac{\lambda^{N+1}}{1-\lambda} M_a$	21.77	41.45	109.31
$\sum_{i=1}^m \left(J_i^0(s_{I_i}) + \frac{\lambda^{N+1}}{1-\lambda} M_i \right)$	51.33	114.98	249.33

Table 4.1 – Minimum, maximum and average values of the dynamic programming functions J_a^0, $J_1^0 + J_2^0$, their difference and the performance guarantees.

the compositional approach since it has to control the water level in tank 2 without knowledge of the control of pump 1. The safety specification for tank 2 thus is realized by using the penalized pump 2 more often. Since $M_1 + M_2 - M_a = 66 \geq 0$, we also know that Corollary 4.9 is satisfied. The guaranteed upper bound on the performance provided for both methods in Theorem 3.10 and 4.5 are given in the bottom of Table 4.1. To see how tight the performance guarantees are, we can compare these values to the worst-case value of the performance criterion:

$$\sum_{k=0}^{+\infty} \lambda^k g(x^k, u^k) = \frac{\max\limits_{x\in[\underline{x},\overline{x}),u\in[\underline{u},\overline{u}]} g(x,u)}{1-\lambda} = 264. \tag{4.25}$$

The average value of the performance guarantee thus is 6.4 times smaller with the centralized method and 2.3 smaller with the compositional approach.

Let us now verify that the performance guarantees from Theorem 3.10 and 4.5 hold. For that, we simulate the behavior of the system controlled with each method and in the same conditions: with a state initialized at the center of the interval and a disturbance input w varying between 0 and $-20\,\text{cm}^3/\text{s}$ as a sine of frequency $1\,\text{rad/s}$. The state variations and control input for each tank are given in Figure 4.5. With the centralized method (red curves), the controller manages to realize the safety specification by using only the pump 1. For the compositional approach (blue curves), the subsystems work without the knowledge of the other control input and without the objective to control the other state. Therefore, in tank 1 we apply the smallest constant control ensuring the safety for x_1 and in tank 2 we need to use the pump 2 to preserve the safety despite the possibly low water inflow from tank 1. The total cost of the controlled trajectories is computed for each method and from any initial state on the trajectory. The minimal, maximal and average values are reported in the first two rows of Table 4.2 and as expected we can see that the centralized method provides much better performances. On the other hand, the compositional method still gives a cost more than twice smaller than the worst-case performance criterion (4.25). In the last two rows of Table 4.2, we verify that Theorem 3.10 and 4.5 are satisfied. We can see that for these simulations, the performance guarantee is really close to the actual performances for the centralized method. For the compositional approach, the loss of information in the subsystems requires taking a larger value for the upper bound of the performances, while the

$k \in \{0, \ldots, 25\}$		$\min_k$	mean_k	$\max_k$
$\sum_{j=0}^{+\infty} \lambda^j g(x^{k+j}, u^{k+j})$	Centralized (C_a^{*X})	31.75	37.74	43.45
	Compositional (C_c^{*X})	26.83	119.70	169.14
$J_a^0(s^k) + \frac{\lambda^{N+1}}{1-\lambda} M_a - \sum_{j=0}^{+\infty} \lambda^j g(x^{k+j}, u^{k+j})$		1.42	4.13	7.27
$\sum_{i=1}^{m} \left(J_i^0(s_{I_i}^k) + \frac{\lambda^{N+1}}{1-\lambda} M_i \right) - \sum_{j=0}^{+\infty} \lambda^j g(x^{k+j}, u^{k+j})$		40.10	63.04	90.70

Table 4.2 – Minimum, maximum and average values of the real cost on the trajectories of the controlled system with each method and the difference with their respective performance guarantees, computed from any starting point on the trajectories.

actual controlled system can reach much smaller costs. △

4.6 Complexity

In Sections 4.4 and 4.5, we have shown that the compositional approach to controller synthesis using symbolic methods provides similar results than the more classical centralized method from Chapter 3: we can synthesize controllers that realize the safety specification (Theorem 4.3) and give some performance guarantees on the controlled trajectories of the original system S (Theorem 4.5). Due to the loss of precision from reasoning on subsystems that only partially cover the global dynamics in (1.1), the results provided by the centralized methods are naturally stronger than the compositional ones: the maximal safe set of S_a solving the safety game contains the one of S_c (Corollary 4.4) and the performance guarantees provided by the centralized method ensures smaller maximal costs (Corollary 4.9). However, these weaker results for the compositional approach come with a possibly greatly reduced computational complexity.

Let us remind the notations $\alpha_x \in \mathbb{N}$ and $\alpha_u \in \mathbb{N}$ from (3.2) and (3.4). For the uniform partition $\mathcal{P}^0$ of the target interval $[\underline{x}, \overline{x}) \subseteq \mathbb{R}^n$ into smaller intervals, α_x represents the number of intervals per dimension, which means that $\mathcal{P}^0$ contains α_x^n symbols. Similarly, the control input interval $[\underline{u}, \overline{u}] \subseteq \mathbb{R}^p$ is discretized into $\alpha_u \geq 2$ values per dimension of the input space, resulting in α_u^p discrete control inputs in U^d.

We focus on the complexity of the main two tasks in these methods: creating finite transition systems corresponding to the abstractions S_a or S_i and solving the dynamic programming algorithm. In Section 3.3, creating the transition system S_a requires, for each symbol $s = [\underline{s}, \overline{s}) \in \mathcal{P}^0$ and control $u \in U^d$, to compute two successors (for $\underline{s}$ and $\overline{s}$) of the sampled system S. For the dynamic programming algorithm applied on S_a, for each step of the time horizon of size N, we have to iterate over all α_x^n symbols and α_u^p inputs, then look among all the possible successors (maximum α_x^n) to compute the cost. For each subsystem S_i, we need to do the same

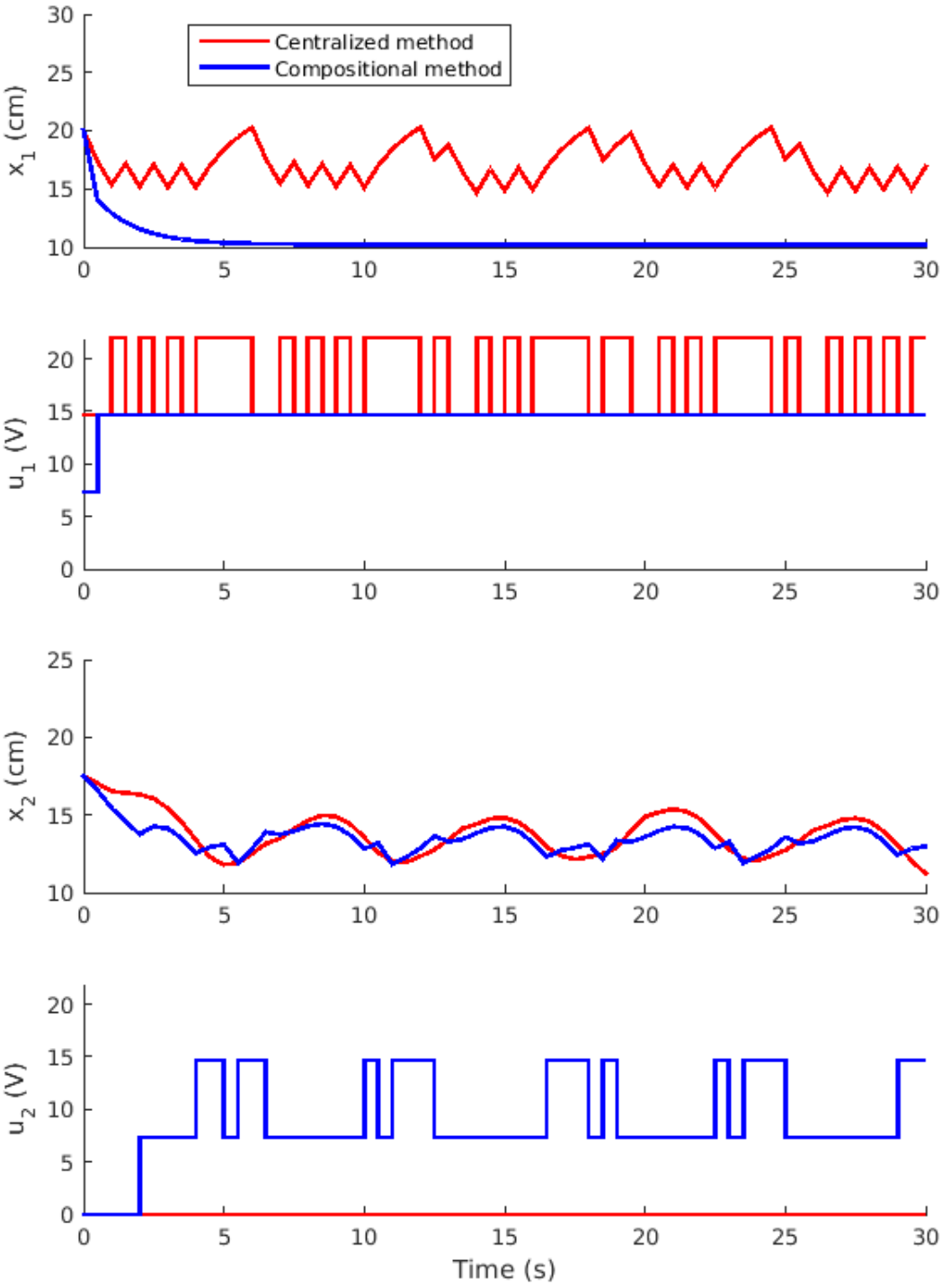


Figure 4.5 – State and control input of each tank, controlled with the centralized method (red curves) and the compositional method (blue curves).

	Method	
	Centralized	Compositional
Abstraction (successors computed)	$2\alpha_x^n\alpha_u^p$	$\sum_{i=1}^{m} 2\alpha_x^{\lvert I_i\rvert}\alpha_u^{\lvert J_i\rvert}$
Dynamic programming (max iterations)	$N\alpha_x^{2n}\alpha_u^p$	$\sum_{i=1}^{m} N\alpha_x^{2\lvert I_i\rvert}\alpha_u^{\lvert J_i\rvert}$

Table 4.3 – Complexity of the abstraction and dynamic programming steps for the centralized and compositional approaches.

but with a lower number of symbols ($\alpha_x^{|I_i|}$) and inputs ($\alpha_u^{|J_i|}$), where $|\cdot|$ represents the cardinality of a set. In Table 4.3, for both the centralized and compositional methods, we give the number of successors of the sampled system S to be computed for the creation of the abstraction S_a or all subsystems S_i and the maximal number of iterations involved in the dynamic programming algorithms.

We can thus see that both the abstraction and dynamic programming steps have an exponential complexity in the dimension of the state space (n for S_a, $|I_i|$ for S_i) and in the dimension of the input space (p for S_a, $|J_i|$ for S_i). The complexity is polynomial in α_x and α_u, the precision level of the state space partition and input space discretization, respectively. Finally, the complexity of the dynamic programming algorithm is linear in the size N of its finite horizon. We should note that the number of iterations provided in Table 4.3 for the dynamic programming algorithm corresponds to a worst-case estimation: most pairs (s, u) usually do not have a transition toward all symbols and the safe set Z_a and the safety controller C_a may also restrict the number of iterations. However, this complexity can significantly be increased from the use of an extended state as described in Remark 3.8. For the compositional method, even though the control objective for subsystem S_i only involves the controlled states (indices I_i^c), the dynamic programming algorithm still has to loop over all the symbol components (indices $I_i \supseteq I_i^c$) as they provide information on the possible successors.

Using a compositional method instead of a centralized one, the complexity can be reduced in two ways. The first one is to increase the number m of subsystems: since $(J_1, \ldots, J_m)$ is a partition of $\{1, \ldots, p\}$, this naturally decreases the factor $\alpha_u^{|J_i|}$. For the same reason, it also decreases the value $|I_i^c|$, but this does not necessarily affect the complexity. Indeed, we can see in Table 4.3 that the complexity actually is influenced by the index sets $I_i = I_i^c \cup I_i^o$ which are not constrained by the number m of subsystems. To reduce the influence of the state on the complexity, we thus need to reduce the precision of the subsystems by decreasing the number of uncontrolled states (indices I_i^o) that are modeled in the subsystems.

Example 4.5. To illustrate the effect of increasing the number of subsystems, consider that we create the centralized abstraction of a model by computing $2\alpha_x^n\alpha_u^p$ successors of the sampled system S. If we use instead a compositional approach with $m = 2$ subsystems that keep the model of the whole state ($I_1 = I_2 = \{1, \ldots, n\}$) but split the control variables into two equal sets ($|J_1| = |J_2| = p/2$), then the compositional

method only needs to compute $2\alpha_x^n \alpha_u^{p/2}$ successors for each subsystems. Hence, due to the exponential complexity in the dimension of the input space, splitting this dimension in halves results in taking the square root of the corresponding complexity (α_u^p). In this example, the overall complexity has been divided by $\alpha_u^{p/2}/2$. △

4.7 Particular cases

Two particular cases can be extracted from the general definition of the decomposition into subsystems described in Section 4.2. The first one is the centralized approach from Chapter 3: if we take $m = 1$, we necessarily obtain $I_1 = I_1^c = \{1, \dots, n\}$, $J_1 = \{1, \dots, p\}$ and $I_1^o = K_1 = L_1 = \emptyset$. From this, A/G Obligation 1 does nothing as all states are observed ($K_1 = \emptyset$) and the unique subsystem S_1 created in Section 4.3 is equal to S_a. A/G Obligation 2 also has no effect since all states are controlled ($I_1^o = \emptyset$). Then with $S_1 = S_a$ the controller synthesis (for safety and performance optimization) is the same as in the centralized method. To sum up, the compositional method presented in this chapter does work for $m = 1$ and gives the same results as the centralized approach from Chapter 3.

A second interesting case that significantly simplifies the controller synthesis is when all the state components modeled in a subsystem are to be controlled. In this case, for each subsystem S_i, we have $I_i^o = \emptyset$ and $I_i = I_i^c$. The creation of the symbolic abstraction S_i is simpler than the general case since all state components are controlled ($I_i^o = \emptyset$) and A/G Obligation 2 has no effect. For the same reason, the restriction on the cost functions g_i in (4.5) disappears. In addition, since $(I_1, \dots, I_m)$ and $(J_1, \dots, J_m)$ are partitions of $\{1, \dots, n\}$ and $\{1, \dots, p\}$ respectively, all the variables and sets involved in two subsystems are defined on disjoints sets of dimensions. This means that all compositions of sets are obtained using the Cartesian product instead of the operator $\Cap$ introduced in (4.1). The last modification is the one stated in Remark 4.7: the composition of the transitions now applies on decoupled systems, which means that the subsystems do not need to synchronize their successors on common dimensions. As a result, we have an equality in Proposition 4.6: $J_c^k(s) = \sum_{i=1}^m J_i^k(s_{I_i})$ and solving the dynamic programming algorithm on each subsystem is equivalent (but computationally much cheaper) to solving it on the composition S_c. Note that the symbolic abstractions S_i computed with this method use a more conservative over-approximation of the reachable set than the general case with the same partition $(I_1^c, \dots, I_m^c)$ since it has no information on the uncontrolled state components ($I_i^o = \emptyset$). This means that the safe set and safety controller will contain less symbols and control inputs. On the other hand, the complexity of this method is reduced compared to the general case. In particular, if we take as many subsystems as state components ($m = n$), the exponential complexity in n and the polynomial complexity in α_x are converted into linear complexities in n and α_x.

Chapter 5

UFAD control in intelligent buildings

The purpose of this chapter is to provide an experimental validation of the control strategies based on robust controlled invariance in Chapter 2 and symbolic control in Chapters 3 and 4. As suggested by the title of this thesis, we are interested in applications to the control of intelligent buildings, also known under various other names such as green, sustainable or smart buildings. Such structures are equipped with sensing and actuation capabilities that allow for an energetically efficient use of the indoor climate control (e.g. light, ventilation, temperature). Here, we focus on the control of the temperature in each room of a building equipped with the UnderFloor Air Distribution (UFAD) solution. Compared to the more traditional ceiling-based ventilation where both the incoming and outgoing air flows are managed in the ceiling *plenum* (a small space common to all rooms and located above a fake ceiling), a UFAD building has two plena: the air is controlled to an appropriate temperature in an underfloor plenum before being sent into each room while the excess of air in each room is pushed into a ceiling plenum through exhausts in the fake ceiling. This application is the initial motivation for the work of this thesis. As stated in Section 1.2, the results presented in Chapters 2 to 4 and the corresponding assumptions required on the system in Chapter 1 come from the generalization of preliminary work focusing on the UFAD building described in the next sections.

This chapter is organized as follows. We first motivate the interest in this application and give an overview of the related work in the field of intelligent buildings in Section 5.1. The experimental UFAD building is described in Section 5.2, where we also present a model for the temperature variations in each room and evaluate it on the real system. Some mathematical properties of this model are given in Section 5.3 to prove that it satisfies all the assumptions in Chapter 1. The control strategies developed in Chapters 2 to 4 are then applied to the experimental building in the next two sections: Section 5.4 for the method based on robust controlled invariant intervals and Section 5.5 for symbolic control. Finally, some concluding remarks comparing these methods are given in Section 5.6.

A first numerical implementation of the results on robust controlled invariance was presented in [MGW13] on a 2-room model inspired by the experimental build-

ing. The experimental validation of these results on the UFAD building was given in [MNGW13, MNGW14], along with an identification procedure to validate the model and obtain its parameters corresponding to the real system. An extension of these results to the robust set stabilization from Section 2.5 with a similar experimental validation appears in [MGWa]. For symbolic control, the centralized method from Chapter 3 was validated on the experimental building in [MGW15]. Lastly, a preliminary version of the compositional approach (the particular case presented in Section 4.7) was illustrated and compare to the centralized approach in a numerical example based on the 2-room UFAD model [MGWb].

5.1 Motivations and related work

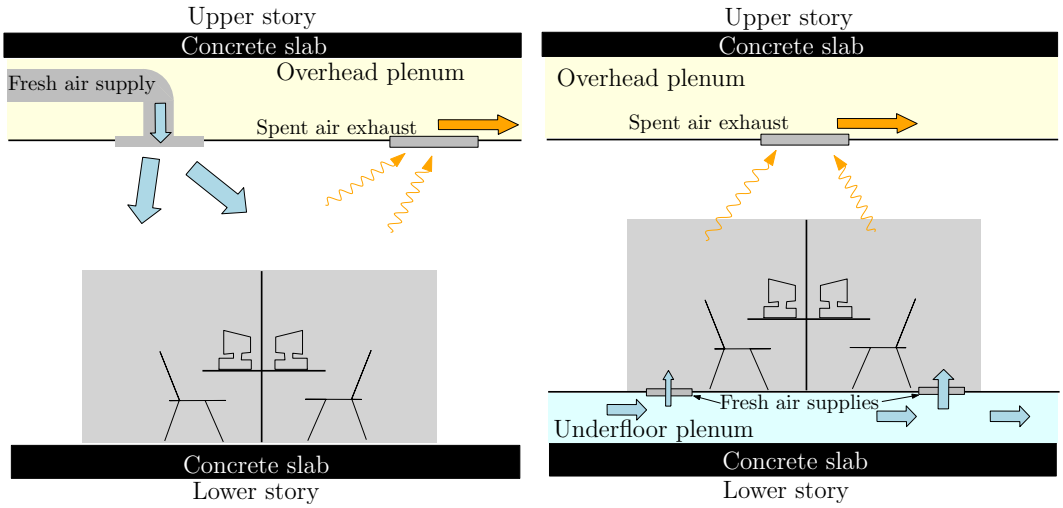
Energy consumption The rapidly growing worldwide energy consumption is a major concern in most countries as it raises numerous challenges on, e.g. the infrastructures for extraction, transformation, storage and transport to provide more energy in response to the growing demand, the exhaustion of the most used resources and the research for sustainable alternatives, or the environmental impact. The global energy consumption is mainly divided between the three major sectors that are industry, transportation and buildings [IEA14]. In particular, buildings represent up to 40% of the total energy consumption in developed countries [PLOP08]. This value is rapidly increasing, not only because of the population growth, but also because of the growing demand of comfort. For example, in the USA, Heating, Ventilating and Air Conditioning systems (HVAC) represent half of the building consumption, hence about 20% of the total energy consumption of the country. These observations have led many countries and regions to consider energy efficiency in buildings as a priority. We could cite for example the European legislation in the Energy Performance of Buildings Directive [EPB02, EPB10] in 2002 and 2010 and the Energy Efficiency Directive [EED12] in 2012.

Intelligent buildings As a result, a significant amount of work has been done in the past decades toward the development of energetically efficient buildings, also called intelligent buildings. The concept of intelligent building was first introduced in the early 1980s and initially only focused on the technological aspect. Although there is no consensus on an official definition of intelligent buildings, it currently covers the aspects of autonomy, comfort, performance and efficiency, adaptability and learning, reduced environmental impact and life cycle cost. In the review [WLW05] are detailed the numerous research topics, both technological and theoretical, related to the development and improvement of intelligent buildings.

HVAC Our focus in this thesis is on the development of innovative control strategies for Heating, Ventilating and Air Conditioning systems (HVAC). Such systems are used to regulate the main components defining the indoor climate such as temperature, ventilation, carbon dioxide levels and humidity. In older buildings, these actions are usually realized separately using, e.g. radiators for heating, fans for ventilation and air conditioning units for cooling. This separation is one of the reasons why such buildings do not enter in the category of intelligent buildings as the absence

of communication and coordination in these actions renders the global regulation particularly inefficient, both in terms of comfort and energy consumption. On the other hand, centralizing the control of all these aspects offers the possibility of great improvements by various methods. This has been the source of numerous research on modeling and simulation of HVAC systems [TH10] as well as on control techniques [MSGT08].

HVAC control The survey paper [MSGT08] categorizes HVAC control methods in three classes. The first class corresponds to traditional methods that are well-known and easy to implement but offer little to no room for energy efficiency. These methods are the on/off control described in the introduction of Chapter 2 (Figure 2.1) to keep the state in an interval and the classical Proportional-Integral-Derivative (PID) controller to have the state follow a setpoint. The class of advanced methods contains controllers offering more flexibility and a first step toward the optimization of some performances. We can thus find in this class methods such as auto-tuning PID [WHZB99], non-linear controller [ASVR99], $\mathcal{H}_\infty$ controller for a robust approach [WDMPB10] or optimal controller [HS95]. Finally, the class of intelligent controllers contains the control methods that can adapt to the Multiple-Input Multiple-Output non-linear and time-varying systems: a better knowledge of the current behavior of the system allows for more freedom in the optimization tasks. In this last class we can find, for example, controllers based on fuzzy logic [HL98] or neural networks [TC98] and model predictive control [OPJ+10]. As all these methods are part of the continuous control theory, our objective in this thesis is to approach the control problem from a different angle by using the symbolic methods described in Chapters 3 and 4. This approach has two main advantages: we can work on a simpler finite model that simulates the behavior of the original system and we can take advantage of the well-established discrete controller synthesis methods such as those in the domain of supervisory control [RW87] or game theory [PPS06].



(a) Overhead ventilation.

(b) UnderFloor Air Distribution.

Figure 5.1 – Comparison of ventilation architectures.

UFAD In this chapter, we are interested in a particular HVAC solution called *UnderFloor Air Distribution* or UFAD. Traditionally, air-conditioning and ventilation are done from an overhead plenum: a space between a fake ceiling and the concrete slab of the floor of the upper story. This means that both the supply of fresh air and the return of spent air are done at the ceiling level which implies a lot of mixing that may create discomfort for the users. In the more recent UFAD solution, the overhead plenum is kept for the return of spent air that naturally rises with warm air, but the air conditioning and the supply of fresh air is placed in an underfloor plenum: a space between a raised floor and the concrete slab of the lower story. This technology has been shown to provide several advantages in terms of user comfort and energy efficiency compared to overhead ventilation [BD03]. In particular, with the supply at the underfloor, the exhaust at the ceiling and the natural rise of warm air, the climate specified by the users can be reached with a much more gentle ventilation, thus increasing the comfort and reducing the energy consumption. The UFAD solution finds two additional advantages when it is integrated in office buildings with an open-plan architecture, where the partitions separating the work stations stop at mid-height and do not go all the way to the ceiling. First, in this type of office buildings an underfloor is usually already existing to hide cables and it can then be easily combined with UFAD technology. Second, with this upper area that is common for the whole floor, the traditional overhead ventilation can only set a single value of the temperature for all users, while the UFAD solution is more flexible since each user can control the opening of the supply outlet on the floor of its station. Both overhead and UFAD solutions are illustrated in Figure 5.1 in the case of an open-plan architecture to illustrate the increased flexibility of the UFAD.

5.2 System description

Our work on UFAD is based on a small-scale experimental building equipped with UnderFloor Air Distribution that is built in the physics department (UFR PhITEM) of University of Grenoble, France. In this section, we thus present the architecture of this UFAD experiment and the associated model for the temperature variations in each room. Then, the model is identified and validated using experimental data and we finally discuss the limitations of the chosen model.

5.2.1 Experimental UFAD building

We consider the small-scale experimental building equipped with UnderFloor Air Distribution pictured in Figure 5.2. This experiment is a PVC box with a volume of approximately one cubic meter. From top to bottom, it is composed of a ceiling plenum, the main central area containing four rooms and an underfloor plenum. Note that unlike Figure 5.1 (b) this building does not have an open-plan architecture and the walls separating rooms go from the raised floor to the fake ceiling. Both the ceiling and underfloor plena are a single space common to all rooms. The rooms are connected by doors that can be controlled in two positions: open or closed. Some halogenic light bulbs are placed in each room to create heat sources that also have two states: on or off.

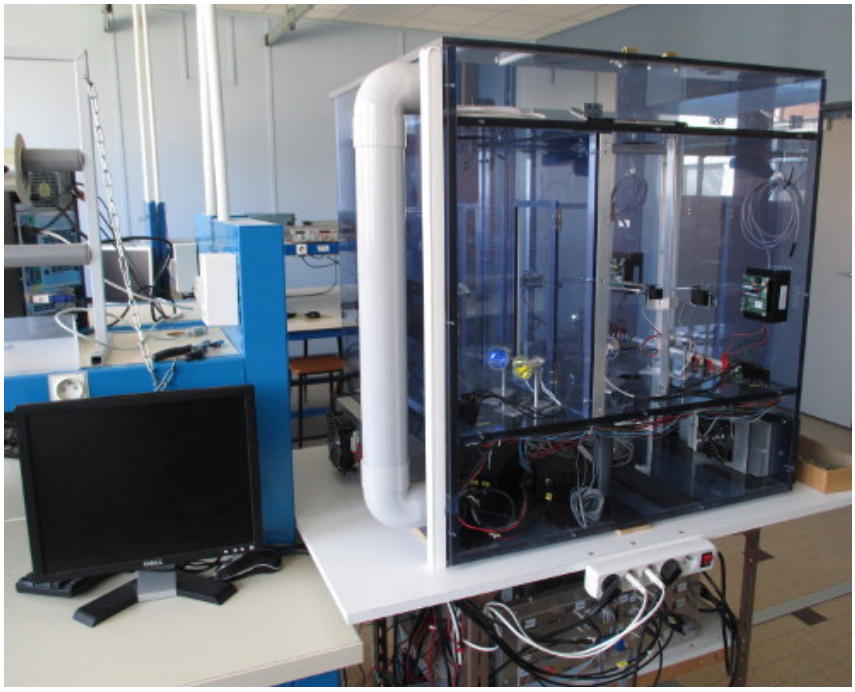


Figure 5.2 – Small-scale experiment of a flat equipped with UFAD.

For a better understanding of its general architecture, we also provide a sketch of this building in Figure 5.3. Our control method focuses on the temperature regulation in the rooms using the active diffusers. The first step is to cool down the air in the underfloor plenum using Peltier coolers. Then, this cold air can be sent into each room with the control fans placed at the level of the raised floor. The excess of air in the room is naturally pushed into the ceiling plenum through an exhaust in the fake ceiling. Finally, another fan sends this spent air back to the underfloor plenum through a return pipe to be cooled down again. Note that the fact that the air is moving in a closed circuit is not a problem since no living person can be in this small-scale flat. However, in an actual building we would need to regularly renew the air to keep an acceptable balance between the levels of oxygen and carbon dioxide.

The building is controlled from a computer with the software LabVIEW^{TM}. The communications with the building go through a CompactRIO, a real-time controller from National Instruments. The building contains six temperature sensors whose values can be read on the computer: one in each room and one in each plenum. We can also control the following actuators: the three Peltier coolers in the underfloor plenum, the four underfloor fans sending cold air into each room, the fan in the return pipe, the opening of the four doors and the lighting of the heat sources.

5.2.2 Model of the temperature variations

There are two components in the control problem. At the building level, we control the air recirculation with the fan in the return pipe and the temperature in the

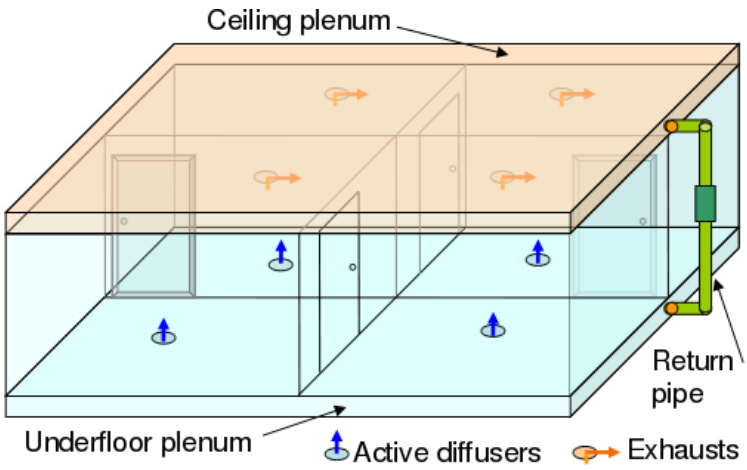


Figure 5.3 – Sketch of the 4-room UFAD building.

underfloor plenum with the Peltier coolers. At the room level, we use the fans in the underfloor to send cold air into each room. Since there is one fan per room and they can be activated separately, we obtain a decentralized control of each room temperature. In this chapter, the controls at the building level are assumed to be set and we focus on achieving the climate regulation in all four rooms.

Hypotheses An initial version of the model of the temperature variations in each room for the experimental flat is given in [WDMPB10]. This model is obtained under the following hypotheses.

(H1) The mass of the air in a room is sufficiently small to neglect its potential energy.

(H2) The speed of the air in a room is sufficiently small to neglect its kinetic energy.

(H3) The speed of the air in a room is sufficiently small to consider the air as incompressible (uniform density: for all room i, ρ_i is equal to the air density ρ).

(H4) The temperature in a room is uniform and its value is the one measured by the unique sensor in the room.

(H5) The air follows the ideal gas law.

The hypothesis (H4) is similar to a lumped model assumption where the variations of the temperature along the spatial dimensions are neglected, thus providing a finite-dimensional model (ordinary differential equations) instead of partial differential equations.

The model for the temperature variations in a room is obtained by combining the mass conservation and energy conservation equations in this room. In what follows, we detail the expression of these two equations and their components (mass flow rates, heat transfers). The formulation of the individual components were validated using the following books and reports [MPS10, Lev02, vdMAB$^+$92]

Mass conservation With the incompressibility hypothesis (H3), the mass conservation equation in room i simply writes as $\sum_k \dot{m}_{k\to i} = \sum_k \dot{m}_{i\to k}$ where $\dot{m}_{k\to i}$ and $\dot{m}_{i\to k}$ are the input and output mass flow rates for room i, respectively. Note that this notation implicitly assumes that both input and output rates are positive. In our system, we have three types of mass flow rates.

- $\dot{m}_{u\to i}$ is the mass flow rate from the underfloor plenum (index u) to room i forced by the corresponding fan. This is always an input mass flow rate for room i.
- $\dot{m}_{i\to c}$ is the mass flow rate from room i to the ceiling plenum (index c), corresponding to the spent air pushed through the exhaust by the fresh air from the underfloor. This is assumed to always be an output mass flow rate for room i.
- $\dot{m}_{d_{ij}}$ is the mass flow rate going through the open door between rooms i and j and can go in either directions.

Let ρ be the density of air, R the air-specific gas constant, A_d the surface of the door opening and T_i the temperature of room i. The direction and value of the mass flow rate going through an open door can thus be given as follows.

Proposition 5.1. *Under hypotheses (H3) and (H5), the net mass flow rate $\dot{m}_{d_{ij}}$ always goes from the warmer to the colder room and its value is given by $\dot{m}_{d_{ij}} = \rho A_d \sqrt{2R|T_i - T_j|}$.*

Proof. With (H5), we can apply the ideal gas law to the volume of room i:

$$P_i = \rho_i R T_i, \tag{5.1}$$

where R is the air-specific gas constant and P_i, ρ_i and T_i are the pressure, density and temperature of the air in the volume V_i. Due to the incompressibility hypothesis (H3), the density is the same for all rooms: $\rho_i = \rho$. Then for two rooms i and j connected by an open door,

$$T_i > T_j \Leftrightarrow P_i > P_j. \tag{5.2}$$

In this case, the mass flow rate $\dot{m}_{d_{ij}}$ thus goes from room i to room j to balance the pressures.

To obtain $\dot{m}_{d_{ij}}$, we consider Bernoulli's principle for incompressible gas (H3):

$$\frac{v_i^2}{2} + g z_i + \frac{P_i}{\rho_i} = \frac{v_j^2}{2} + g z_j + \frac{P_j}{\rho_j}, \tag{5.3}$$

where v is the speed of the air, g is the gravitational constant and z is the elevation. With (H1) we can neglect the potential energy terms gz_i and gz_j. Hypothesis (H3) also gives $\rho_i = \rho_j = \rho$. In the case $T_i > T_j$, we use (H2) to neglect the kinetic energy the air in the warmer room ($v_i^2/2$) and we look for the speed $v = v_j$ in the colder room induced by the pressure difference in (5.2). Combining (5.1) with (5.3) in these conditions, we obtain:

$$v = \sqrt{2\frac{P_i - P_j}{\rho}} = \sqrt{2R(T_i - T_j)}.$$

With A_d denoting the area of the door, the mass flow rate is

$$\dot{m}_{d_{ij}} = \rho A_d v = \rho A_d \sqrt{2R(T_i - T_j)}. \qquad \square$$

Using the sign function sign : $\mathbb{R} \to \{-1, 1\}$ to represent the direction of the mass flow rates at open doors, the mass conservation equation in room i can thus be written as follows:

$$\dot{m}_{u\to i} - \dot{m}_{i\to c} + \sum_{j\in\mathcal{N}_i} \delta_{d_{ij}} \operatorname{sign}(T_j - T_i)\dot{m}_{d_{ij}} = 0, \tag{5.4}$$

where $\mathcal{N}_i$ is the set of room indices having a common door with room i and $\delta_{d_{ij}} \in \{0, 1\}$ is the state (closed/open) of the door between rooms i and j.

Energy conservation The energy conservation in a room i is described by the first law of thermodynamics:

$$\frac{dE_i}{dt} = \dot{Q}_{cond_i} + \dot{Q}_{rad_i} + \sum_k h_k \dot{m}_{k\to i} - \sum_k h_i \dot{m}_{i\to k}. \tag{5.5}$$

In this equation, E_i represents the energy of the room i. With hypotheses (H1) and (H2), the potential and kinetic energies are neglected and E_i can be approximated by the internal energy $E_i = \rho V_i C_v T_i$, where ρ is the density of air, V_i the volume of the room, C_v the constant volume specific heat and T_i the room temperature. In the right-hand side of (5.5), we can see that variations of the energy are induced by four types of heat transfers:

- $\dot{Q}_{cond_i}$ is the thermal conduction through the walls of the room;
- $\dot{Q}_{rad_i}$ is the radiation from heat sources in the room;
- $h_j \dot{m}_{j\to k}$ is the heat transfer induced by the mass flow rate $\dot{m}_{j\to k}$, where h_j is the enthalpy of the room from which the mass flow rate is coming and can be approximated as $h_j = C_p T_j$ using the ideal gas hypothesis (H5), with the specific heat at constant pressure C_p.

Note that as in the mass conservation equation (5.4), the mass flow rates that appears in (5.5) are positive, which explains the positive sign before the incoming rates and the negative sign before the outgoing rates. An additional heat transfer $\dot{Q}_{conv_i}$ could be added in (5.5) to represent the convection between the air and the walls. Due to some undesirable behaviors that would be induced by this term (e.g. convection coefficient that depends on the ventilation), we decide to discard it for now and we discuss in Section 5.2.4 how it can be included in the model.

Using the previously introduced set $\mathcal{N}_i$ denoting the room indices having a common door with room i, we extend it into $\mathcal{N}_i^* = \mathcal{N}_i \cup \{u, c, o\}$ with the indices u, c and o representing the underfloor plenum, the ceiling plenum and the outside of the building, respectively. $\mathcal{N}_i^*$ thus represents all spaces in contact with room i through a wall: according to Figure 5.3, two rooms, both plena and the outside.

Using Fourier's law, the thermal conduction through a wall is proportional to the temperature difference, which gives for room i:

$$\dot{Q}_{cond_i} = \sum_{j \in \mathcal{N}_i^*} \frac{k_{ij} A_{ij}}{\Delta_{ij}} (T_j - T_i), \tag{5.6}$$

where k_{ij}, A_{ij} and Δ_{ij} are respectively the conductivity, surface and thickness of the wall separating room i and the space of index j. As expected, the heat transfer is positive if the temperature T_j of the neighbor space is greater than the room temperature T_i.

Let $\delta_{s_i} \in \{0, 1\}$ denote the discrete state (off/on) of a heat source of temperature T_{s_i}. The radiative heat transfer from this source to the air in room i is given by:

$$\dot{Q}_{rad_i} = \delta_{s_i} \varepsilon_{s_i} \sigma A_{s_i} (T_{s_i}^4 - T_i^4), \tag{5.7}$$

where ε_{s_i} and A_{s_i} are the emissivity and surface area of the heat source and σ is the Stephan-Boltzmann constant.

Using the known direction of the mass flow rates affecting room i from the previous paragraph, the energy conservation equation (5.5) thus can be written as:

$$\begin{aligned} \rho V_i C_v \frac{dT_i}{dt} = & \sum_{j \in \mathcal{N}_i^*} \frac{k_{ij} A_{ij}}{\Delta_{ij}} (T_j - T_i) + \delta_{s_i} \varepsilon_{s_i} \sigma A_{s_i} (T_{s_i}^4 - T_i^4) \\ & + C_p T_u \dot{m}_{u \to i} - C_p T_i \dot{m}_{i \to c} \\ & + \sum_{j \in \mathcal{N}_i} C_p \max(T_i, T_j) \delta_{d_{ij}} \operatorname{sign}(T_j - T_i) \dot{m}_{d_{ij}}. \end{aligned} \tag{5.8}$$

For the last line of (5.8) when the door d_{ij} is open ($\delta_{d_{ij}} = 1$), if $T_j > T_i$ we have a mass flow rate going from the warmer room j to the colder room i with an associated heat transfer $C_p T_j \dot{m}_{d_{ij}} > 0$. Similarly when $T_j < T_i$, the mass flow rate goes in the opposite direction and the heat transfer is negative: $-C_p T_i \dot{m}_{d_{ij}} < 0$.

Final model As the temperatures are measured, $\dot{m}_{d_{ij}}$ is known from Proposition 5.1 and $\dot{m}_{u \to i}$ is linked to our control input. The only unknown variable in (5.8) is the mass flow rate $\dot{m}_{i \to c}$ going from room i to the ceiling plenum. We thus replace it in the energy conservation equation (5.8) by its expression obtained from the mass conservation equation (5.4):

$$\begin{aligned} \rho V_i C_v \frac{dT_i}{dt} = & \sum_{j \in \mathcal{N}_i^*} \frac{k_{ij} A_{ij}}{\Delta_{ij}} (T_j - T_i) + \delta_{s_i} \varepsilon_{s_i} \sigma A_{s_i} (T_{s_i}^4 - T_i^4) + C_p \dot{m}_{u \to i} (T_u - T_i) \\ & + \sum_{j \in \mathcal{N}_i} C_p \delta_{d_{ij}} \operatorname{sign}(T_j - T_i) \dot{m}_{d_{ij}} (\max(T_i, T_j) - T_i). \end{aligned}$$

Since we assumed that the mass flow rate between a room and the ceiling plenum always goes up, it is associated with the temperature of the room T_i in the energy conservation equation (5.8). The heat transfer related to the underfloor fan is thus proportional to the temperature difference between the room and the underfloor plenum. For the heat transfer linked to an open door, it naturally depends on which

room is the warmer: if $T_j > T_i$, we obtain $C_p \dot{m}_{d_{ij}}(T_j - T_i)$, but we obtain 0 when $T_j < T_i$. This means that an open door with outgoing air flow ($T_j < T_i$) as no more effect on the temperature variations in the considered room than a closed door. This can be explained by the assumption that $\dot{m}_{i\to c}$ always goes toward the ceiling plenum. Indeed, whether the door d_{ij} is open with $T_j < T_i$ or closed, $\dot{m}_{u\to i}$ is the only incoming mass flow rate, therefore with the mass conservation equation (5.4), $\dot{m}_{u\to i}$ also is the value of the total outgoing mass flow rate. Since the energy term linked to all outgoing mass flow rates involves the enthalpy $h_i = C_p T_i$ of room i, the general definition of the energy conservation equation (5.5) necessarily writes as:

$$\frac{dE_i}{dt} = \dot{Q}_{cond_i} + \dot{Q}_{rad_i} + C_p T_u \dot{m}_{u\to i} - C_p T_i \dot{m}_{u\to i},$$

which means that it is not important whether the energy $-C_p T_i \dot{m}_{u\to i}$ leaves room i by the door or the ceiling exhaust (or any intermediate combination) as long as it does leave the room (mass conservation). Thus in this case ($T_j < T_i$), the state of door d_{ij} does not affect the temperature variations of room i. Note that this assumption is reasonable since when such ventilation systems are built, we usually want to prevent (using difference of pressure or sufficient ventilation) the spent air in the ceiling plenum from coming back into the room.

When we replace the mass flow rate through a door by its expression $\dot{m}_{d_{ij}} = \rho A_d \sqrt{2R|T_i - T_j|}$ from Proposition 5.1, we obtain the final formulation of the model:

$$\begin{aligned}\rho V_i C_v \frac{dT_i}{dt} = &\sum_{j\in\mathcal{N}_i^*} \frac{k_{ij}A_{ij}}{\Delta_{ij}}(T_j - T_i) + \delta_{s_i}\varepsilon_{s_i}\sigma A_{s_i}(T_{s_i}^4 - T_i^4) \\ &+ C_p \dot{m}_{u\to i}(T_u - T_i) + \sum_{j\in\mathcal{N}_i} \delta_{d_{ij}} C_p \rho A_d \sqrt{2R}\max(0, T_j - T_i)^{3/2}.\end{aligned} \tag{5.9}$$

Dry friction Although in simulations, we can easily consider the mass flow rate $\dot{m}_{u\to i}$ as the control input of (5.9), in reality we do not have a direct control of this flow. In the small-scale experiment from Figure 5.2, the four underfloor fans are regulated by applying a voltage. In addition, the fans are significantly affected by dry friction. As illustrated in Figure 5.4, for voltages V_i smaller than a threshold V_i^*, the dry friction prevents the fan from moving and the mass flow rate $\dot{m}_{u\to i}$ is equal to zero. What happens for $V_i > V_i^*$ is unknown to us since we have no tool to measure actual mass flow rates. Therefore, we assume that the relation between the voltage and the mass flow rate is affine as in Figure 5.4. Let $\overline{\dot{m}_{u\to i}}$ denote the value of the mass flow rate obtained with the maximal voltage command $\overline{V_i}$. The mass flow rate created by the underfloor fan can thus be expressed in terms of its voltage input as follows:

$$\dot{m}_{u\to i}(V_i) = \begin{cases} 0 & \text{if } V_i \le V_i^*, \\ \dfrac{\overline{\dot{m}_{u\to i}}}{\overline{V_i} - V_i^*}(V_i - V_i^*) & \text{if } V_i > V_i^*. \end{cases} \tag{5.10}$$

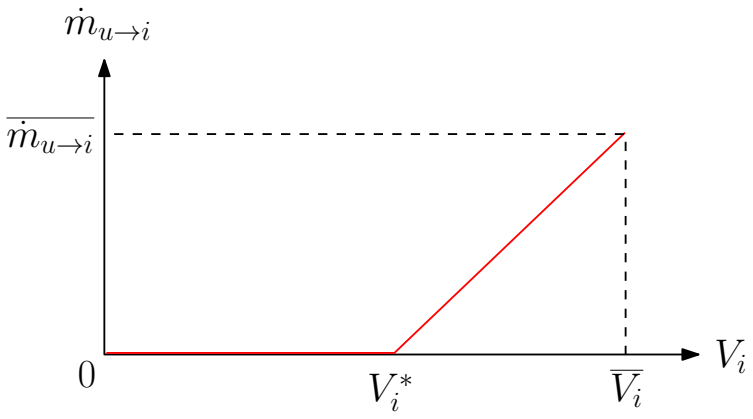


Figure 5.4 – Dry friction in the command of an underfloor fan.

5.2.3 Evaluation of the model

Since the control methods from Chapters 2 to 4 are based on the model of the process to control, we need to verify that the small-scale experiment can be represented by a model of the form (5.9) and to identify the value of the unknown parameters corresponding to this building. In theory, most of the parameters involved in (5.9) are known or can be estimated:

- V_i, A_{ij}, Δ_{ij}, A_{s_i} and A_d are geometrical properties that can be measured on the building;
- ρ, C_v, σ, C_p and R are known physical constants;
- k_{ij}, ε_{s_i}, T_{s_i}, $\overline{\dot{m}_{u\to i}}$ and V_i^* may be given in the technical characteristics provided by the manufacturers of the PVC used for the walls, the lamps and the fans.

Only the third category might not be fully known. However, in Section 5.2.2, many approximations have been made to obtain a simpler model (5.9), but it also results in having a less precise model. To compensate these simplifications, we choose the more flexible gray-box identification procedure where we impose the general form of the dynamics (5.9) but all constant parameters are aggregated into one value per heat transfer. The identification objective is thus to find the value of these abstract parameters such that the following model matches the measured behavior of the experiment:

$$\begin{aligned}\frac{dT_i}{dt} = &\sum_{j\in\mathcal{N}_i^*} a_{ij}(T_j - T_i) + \delta_{s_i} b_i (T_{s_i}^4 - T_i^4) \\ &+ c_i \max\left(0, \frac{V_i - V_i^*}{\overline{V_i} - V_i^*}\right)(T_u - T_i) + \sum_{j\in\mathcal{N}_i} \delta_{d_{ij}} d_{ij} \max(0, T_j - T_i)^{3/2}.\end{aligned} \tag{5.11}$$

To keep the model relatively simple and decrease the complexity of the identification procedure, we assume that the voltage V_i and the mass flow rate $\dot{m}_{u\to i}$ have an affine relation with dry friction, as illustrated in Figure 5.4. This has two consequences.

Room i	1	2	3	4
Friction threshold V_i^*	2.5 V	3 V	3 V	2.5 V

Table 5.1 – Values of the dry friction thresholds for the experimental building.

Firstly, after estimating the values V_i^* of the dry friction, the identification can focus on the extremal values 0 and $\overline{\dot{m}_{u\to i}}$ of the mass flow rate, or equivalently, the extremal values 0 and $\overline{V_i}$ of the voltage. Secondly, the maximal value $\overline{\dot{m}_{u\to i}}$ of the mass flow rate that appears in (5.10) can be included in the parameter c_i to be identified in (5.11).

Voltage input The value of the dry friction thresholds V_i^* are estimated visually by slowly incrementing the voltage V_i until the corresponding fan starts moving. The obtained values for the four underfloor fans are given in Table 5.1. Initial observations on the experimental building showed that for large values of the input voltage, the resulting mass flow rate creates behaviors that significantly deviate from those that can be described by the model (5.11). To avoid these behaviors and preserve the structure of this model (and more importantly the mathematical properties proven in Section 5.3), we restrict our tests to a maximal voltage $\overline{V} = 6\,\mathrm{V}$ (while the maximal range physically allowed is 12 V). As a result, this gives a significant importance to the dry friction and it can also explain why in some of the control results discussed in the next sections, the controllers may not be able to reach sufficiently low room temperatures.

Identification procedure The experimental data gathered for the following identification procedure are obtained when the outside temperature T_o is varying around 30 °C and the temperature of the underfloor plenum is regulated at 17 °C using a PID controller. The conditions of each experiment are mainly defined by the discrete state of the doors (open or closed), the discrete state of the heat sources (on or off) and the discrete state of the underfloor fans (maximal voltage $\overline{V} = 6\,\mathrm{V}$ or off, as described in the previous paragraph). Our aim is to quantify the heat transfers involved in (5.11) due to conduction in the walls, radiation of heat sources and exchange of air flows. For each room, we thus aim to cover all the main behaviors, both separately and combined, using the experiments summarized as follows.

- Radiation: a lamp is turned on, we wait for an equilibrium then turn it off.
- Ventilation: a fan is turned on, we wait for an equilibrium then turn it off.
- Door and lamp: a lamp is turned on in a room to create a temperature gradient, then one of its doors is opened.
- Door and fan: a fan is turned on in a room to create a temperature gradient, then one of its doors is opened.
- Lamp and fan: alternatively turn on and off the fan and lamp of the same room to cover all operating conditions.

Room i	1	2	3	4
$a_{i,1}$		7.60×10^{-5}		1.09×10^{-4}
$a_{i,2}$	2.85×10^{-4}		1.79×10^{-4}	
$a_{i,3}$		1.89×10^{-4}		1.07×10^{-4}
$a_{i,4}$	2.47×10^{-4}		3.81×10^{-4}	
$a_{i,u}$	7.36×10^{-5}	7.02×10^{-5}	3.45×10^{-5}	3.26×10^{-5}
$a_{i,c}$	9.27×10^{-5}	2.42×10^{-4}	3.21×10^{-8}	1.73×10^{-4}
$a_{i,o}$	5.78×10^{-4}	6.21×10^{-4}	5.64×10^{-4}	5.99×10^{-4}
b_i	3.12×10^{-17}	2.55×10^{-16}	8.57×10^{-13}	3.57×10^{-17}
T_{s_i}	3.73×10^{3}	1.78×10^{3}	3.80×10^{2}	3.93×10^{3}
c_i	2.12×10^{-3}	1.88×10^{-3}	3.05×10^{-3}	1.40×10^{-3}
$d_{i,1}$		9.26×10^{-4}		2.72×10^{-4}
$d_{i,2}$	1.86×10^{-4}		2.57×10^{-4}	
$d_{i,3}$		8.11×10^{-4}		6.86×10^{-4}
$d_{i,4}$	7.55×10^{-4}		1.98×10^{-8}	

Table 5.2 – Identified parameters of the gray-box model (5.11).

There is no specific experiment for open doors only as nothing would happen if we are already at an equilibrium. Note also that the conduction is naturally included in all tests. Approximately 16 hours of data were recorded to perform all these experiments in each room.

Using the gray-box identification method, the global model has a total of 40 unknown parameters that need to be identified. Indeed, according to the sketch of the building in Figure 5.3, each room described by (5.11) has 10 parameters: 5 for the conduction (a_{ij}) with both its neighbor rooms, both plena and the outside; 2 for radiation (b_i and T_{s_i}); 1 for the ventilation (c_i); and 2 for the doors with its neighbor rooms (d_{ij}). The optimization problem is solved using a least-squares algorithm initialized with a set of values based on known physical parameters and observations. The resulting values of the identified model (5.11) are given in Table 5.2. The comparison with the theoretical values of these parameters according to the physical model (5.9) is not provided here as it presents a significant mismatch: the theoretical values are between 10 and 10^9 times greater than the identified parameters. Some reasons possibly explaining this mismatch are discussed in Section 5.2.4.

Evaluation The identified model (5.11) with the values in Table 5.2 is evaluated on an experimental scenario not included in the data set used for the identification. Starting with all lamps and fans off and all doors closed, the switching scenario of the lamps, fans and doors for this experiment is as follows:

$t = 150\,\text{s}$, lamp 1 and fan 3 on;
$t = 570\,\text{s}$, lamp 3 on, door $1-4$ open;

$t = 810\,\text{s}$, fan 4 on;
$t = 930\,\text{s}$, lamp 3 off, door $1-2$ open;
$t = 1050\,\text{s}$, door $1-4$ closed.

We can notice that this scenario covers all the main heat transfers and their combinations:

- Conduction alone: e.g. rooms 2 and 4 between 150 s and 570 s (with warmer room 1 and colder room 3);
- Radiation: e.g. room 1 between 150 s and 570 s;
- Ventilation: e.g. room 3 between 150 s and 570 s;
- Open door: e.g. room 4 between 570 s and 810 s;
- Radiation and ventilation: e.g. room 3 between 570 s and 930 s;
- Radiation and open door: e.g. room 1 between 570 s and the end;
- Ventilation and open door: e.g. room 4 between 810 s and 1050 s.

In Figure 5.5, for each room we give the graph of the experimental measurements (blue curves) corresponding to this switching scenario and compare it to the theoretical behavior (red curves) of the gray-box model (5.11) with the identified parameters from Table 5.2. The vertical black lines correspond to the transitions in the scenario: plain lines when the switching is linked to the room, dashed otherwise. On this data set of 1211 points (one measure per second), the mean squared error between the model and the measurements is 0.18 with a standard deviation of 0.42. We can see that the identified model fits the experimental data relatively well even though there are some slight variations. The main differences appear in two conditions: when a fan is active in a room with an open door, or when a fan and a lamp are active in the same room.

When we look at the top graph of Figure 5.5, we can see that between 810 s and 1050 s, the door between rooms 1 and 4 is open while the ventilation of room 4 is active. The model (5.11) says that in such conditions, the open door should have no effect on the dynamics of the warmer room: room 1. In reality, we can see that the ventilation in room 4 creates some air circulation (convective effect) that also affects room 1.

The models of the heat transfers for radiation alone (room 1 between 150 s and 570 s) and the ventilation alone (room 3 between 150 s and 570 s) fit well the experimental data. However, when they are combined, there are some unmodeled behaviors due to delays for the heat source to reach its theoretical temperature (T_{s_i} when the lamp is on, T_i when it is off). Both delays can be observed in room 3, where the ventilation is always active after 150 s. At 570 s, the lamp is turned on but it does not immediately reaches its final temperature due to the ventilation cooling it down, which thus results in a smaller heat transfer between 570 s and 930 s. At 930 s, the lamp is turned off but its remaining heat combined with the ventilation continues to warm up the room, while the model assumes that the heat source has no effect.

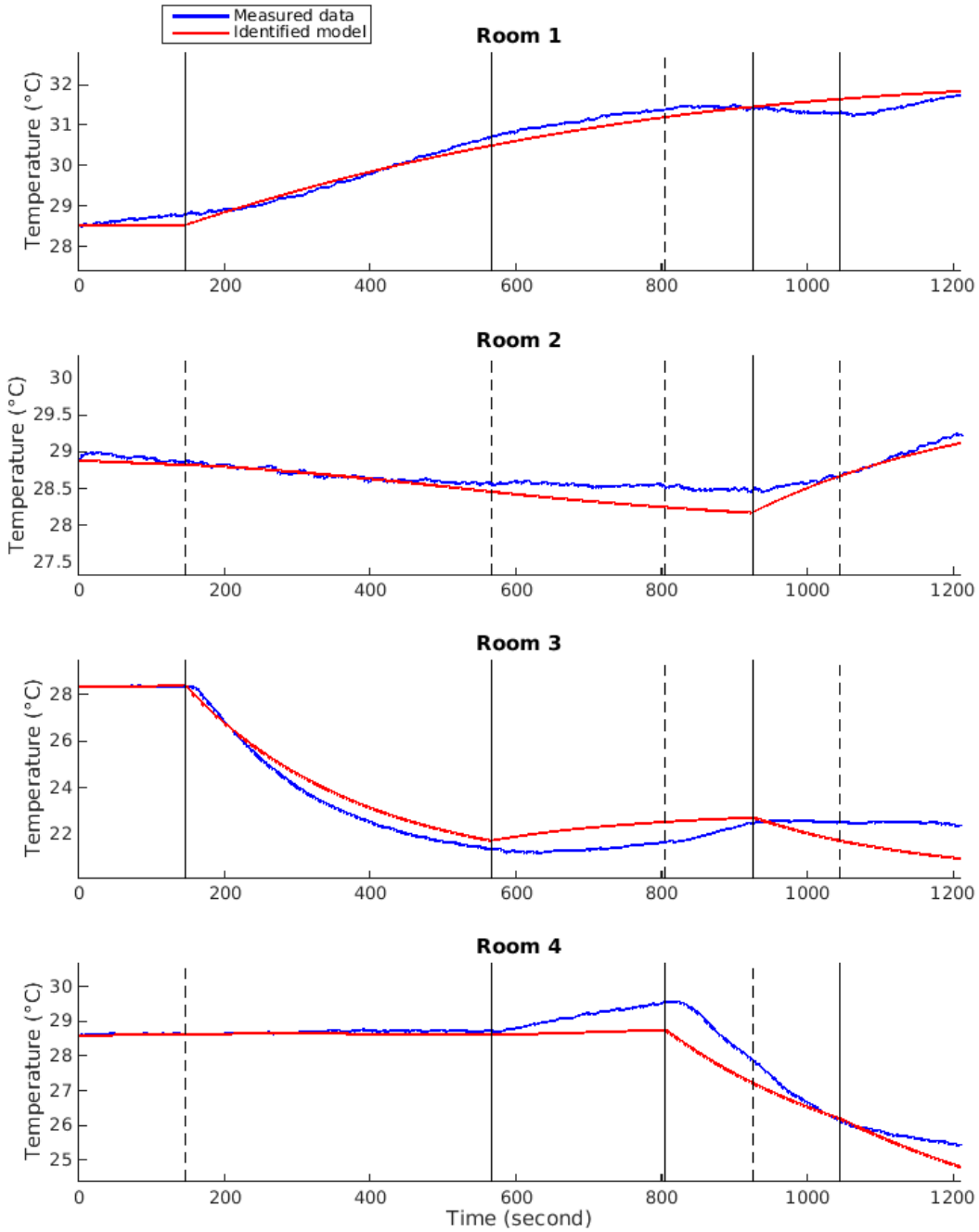


Figure 5.5 – Comparison between the identified model (red) and the experimental measurements (blue) for the evaluation scenario (events represented by vertical plain lines when linked to the room, dashed lines otherwise).

Despite these differences between the identified model (5.11) and the real behavior of the experimental building, we consider this model to fit sufficiently well the experimental data to be used for our control applications in Sections 5.4 and 5.5. The performance limitations of our model and some leads for its improvement are proposed in the next section.

5.2.4 Limitations and possible improvements

Uniform temperature Firstly, the uniformity of the temperature in each room is obviously not realistic: it is well known that there is a stratification where the warmer air goes up. This stratification is reduced in our small(scale experiment, where the fan actuation induces turbulent mixing in our relatively small control volume. In addition, considering the spatial variations in the model results in an infinite dimensional system (partial differential equation) which cannot be used for the control methods presented in this thesis. On the other hand, without necessarily assuming the uniformity of the temperature, it seems natural to consider a single value of the room temperature corresponding to the height of the upper body of the users.

Flow directions From Figure 5.5, we clearly see that the model of the heat transfer linked to the air flow going through a door is not true on the real system. This may have two causes. The first one is the result from Proposition 5.1 stating that under some assumptions, the air flow is unidirectional from the warmer to the colder room. For example, it is possible that the surface area of the open door is too large for Bernoulli's principle to apply as we did in Proposition 5.1 with the same speed of the air at all heights. A more in-depth study on how to apply Bernoulli's principle to compute the air flow through large openings can be found in [vdMAB$^+$92].

The second possible cause is on the reasoning that led the heat transfer linked to the air flow at the door to disappear from the energy conservation equation (5.8) of the warmer room after using the mass conservation equation (5.4). This comes from the assumption that the mass flow rate between the room and the ceiling plenum always goes to the ceiling (hence, the associated heat transfer always uses the temperature of the room). While this might be enforced in a real building by always applying a sufficient ventilation or having a pressure difference between the room and the ceiling plenum, it is not the case in our small-scale experiment: if a door is open and the ventilation of the warmer room is off, the mass conservation equation should give a mass flow rate from the ceiling to the room.

More generally, a more accurate model could probably be obtained if we do not consider only the mass conservation in the four rooms, but also in both the underfloor plenum and the ceiling plenum. The obtained model would be slightly more complicated, but the main problem is that it may not satisfy some important mathematical properties such as the monotonicity. Indeed, for any assumption on the air flow directions, combining the mass conservation and energy conservation equations always results in heat transfers linked to the mass flow rates of the form $\dot{m}_{i\to j}(T_i - T_j)$. If for example we consider a mass flow rate $\dot{m}_{c\to i}$ from the ceiling plenum to a room i, the heat transfer $\dot{m}_{c\to i}(T_c - T_i)$ would have either a positive

or a negative effect on the variations of T_i depending whether $T_c > T_i$ or $T_c < T_i$: this is not a monotone behavior. The simple though incomplete model (5.11) is thus kept to preserve this property.

Conduction and convection In the physical model (5.9), the control volume considered in the first law of thermodynamics only contains the air of the room. From a comparison between the theoretical values in this model and the identified parameters in Table 5.2, it seems more accurate to also consider half of the volume of its six walls in this control volume to be able to consider the heat exchanges at the interface of two control volumes. This addition significantly changes the average value of the density of the control volume (and most of the other air-specific constants considered in (5.9)). This was one of the main reasons to choose a gray-box identification method. An alternative solution would be to model the central temperature of each wall as a state variable and apply the energy conservation equation to the wall. A drawback of this solution is that we have no access to measurements of these temperature, thus rendering the identification task more difficult.

Related to this last solution, we can note that heat transfers from convection are not modeled in (5.9) to represent the exchanges between the air and the walls (while conduction should only describe the exchanges inside a wall). Convection can be added to the model along thermal conduction using an electrical analogy: the conduction or convection heat flux is equivalent to a current, the temperature difference to a difference of potential and the constant factor to the thermal resistance. Hence, for a wall of width Δ, surface A and conductivity k, the conduction $\dot{Q}_{cond} = (T_i - T_j)kA/\Delta$ gives the thermal resistance $R_{cond} = \Delta/(kA)$. Similarly, the convection between a solid of surface A and a fluid of convection coefficient h is given by $\dot{Q}_{conv} = (T_i - T_j)hA$ (Newton's law), which results in a thermal resistance $R_{conv} = 1/(hA)$. If we consider the example in Figure 5.6, the total heat transfer between rooms 1 and 2 through the wall can be obtained by placing the resistances in series and taking the total resistance:

$$\dot{Q}_{total} = \frac{T_1 - T_2}{R_{total}} = \frac{T_1 - T_2}{\frac{1}{h_1 A} + \frac{\Delta}{kA} + \frac{1}{h_2 A}}.$$

Note that the convection coefficient h can take a wide range of values depending on the ventilation: from 5 for the slowest natural convection up to 250 for forced convection.

Radiation As stated at the end of the previous section, we can see in Figure 5.5 that switching the state of a lamp in a room where the ventilation is active adds a delay in the radiative heat transfer. We could thus try to include this delay in the model (5.11). However, we also see in Figure 5.5 that when the lamp is on and without ventilation (room 1 after 150 s), the model of the radiative heat transfer matches the experimental data. The delay that we add should thus probably depend on the mass flow rate.

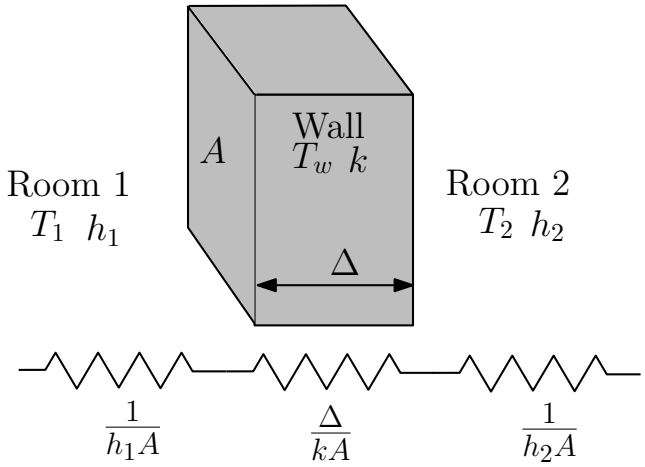


Figure 5.6 – Electrical analogy for thermal conduction and convection.

Control input In the identification procedure, we considered that the mass flow rate corresponding to our control input (the underfloor fan) could only take two values (0 and the mass flow rate induced by the maximal voltage $\overline{V} = 6\,\mathrm{V}$), while we assumed that for intermediate values the relation between the mass flow rate and the voltage is affine with dry friction. A possible improvement of our model is to identify the real function $\dot{m}_{u\to i}(V_i)$ to obtain a more accurate model for intermediate values of the voltage. In addition, it is possible that the dry friction threshold actually has two different values: one preventing the fan from moving for small values of the voltage and a smaller one stopping the fan when its speed decreases.

Conclusion To summarize this section, we have shown that the chosen model has several limitations and could be improved on various aspects. Nevertheless, we have seen in Section 5.2.3 that our model still matches relatively well the experimental behavior. In addition, one of the main challenge in modeling a system or modifying an existing model is to preserve the mathematical properties (studied in the next section) needed later to develop control strategies based on this model. For these reasons, we choose to keep this simplified model (5.11) with the identified parameters in Table 5.2.

5.3 Model properties

To apply the control strategies and other results from Chapters 2 to 4 to the model created in Section 5.2, we need to verify that this model satisfies the assumptions presented in Chapter 1 (Section 1.2). Let us remind the general form of the consid-

ered model for the temperature variations in a room of index i:

$$\frac{dT_i}{dt} = \sum_{j\in\mathcal{N}_i^*} a_{ij}(T_j - T_i) + \delta_{s_i} b_i (T_{s_i}^4 - T_i^4) \tag{5.11}$$
$$+ c_i \max\left(0, \frac{V_i - V_i^*}{\overline{V} - V_i^*}\right)(T_u - T_i) + \sum_{j\in\mathcal{N}_i} \delta_{d_{ij}} d_{ij} \max(0, T_j - T_i)^{3/2},$$

where $\mathcal{N}_i \subseteq \{1,2,3,4\}$ is the set of neighbor rooms for room i, $\mathcal{N}_i^* = \mathcal{N}_i \cup \{u, c, o\}$ includes all neighbor spaces (rooms, plena and outside) and the constant parameters a_{ij}, b_i, c_i and d_{ij} are positive. We can then describe the dynamics of the whole system (four rooms) similarly to (1.1):

$$\dot{T} = f(T, u, w, \delta), \tag{5.12}$$

where T, u, w and δ are defined as follows:

- $T \in \mathbb{R}^4$ is the state of the system, containing the temperature of each room;
- $u \in [-\overline{V}, 0]^4 \subseteq \mathbb{R}^4$ is the control input related to the ventilation of the underfloor fans, with $u_i = -V_i$ for all $i \in \{1,2,3,4\}$. This choice is explained in Section 5.3.1;
- $w = [T_u, T_c, T_o] \in \mathbb{R}^3$ gathers the exogenous temperatures (underfloor, ceiling and outside) considered as continuous disturbances in (5.12);
- $\delta = [\delta_{s_1}, \delta_{s_2}, \delta_{s_3}, \delta_{s_4}, \delta_{d_{12}}, \delta_{d_{23}}, \delta_{d_{34}}, \delta_{d_{41}}] \in \{0,1\}^8$ contains 8 binary disturbances: the state of the heat source in each room and the state of the 4 doors between the rooms.

We can clearly see in (5.11) that each control input $u_i = -V_i$ has only a direct influence on the state T_i of the same room. The local control property from Definition 1.12 and Assumption 2 is thus satisfied.

5.3.1 Monotonicity

Since $3/2 > 1$, the function $x \longmapsto max(0,x)^{3/2}$ is continuously differentiable on $\mathbb{R}$ and, as a result, the vector field f in (5.12) is continuously differentiable in T and w. Using Proposition 1.7, we can prove that (5.12) is cooperative with respect to its state T and continuous disturbance w by computing the corresponding partial derivative of the vector field f. For $i, j \in \{1,2,3,4\}$ and $j \neq i$, the partial derivative with respect to the state components are:

$$\frac{\partial f_i}{\partial T_j}(T, u, w, \delta) = a_{ij} + \delta_{d_{ij}} d_{ij} \frac{3}{2} \max(0, T_j - T_i)^{1/2} > 0$$

if $j \in \mathcal{N}_i$ and 0 otherwise. For the exogenous temperatures, we have:

$$\frac{\partial f_i}{\partial T_u} = a_{iu} + c_i \max\left(0, \frac{V_i - V_i^*}{\overline{V} - V_i^*}\right) > 0 \ ; \ \frac{\partial f_i}{\partial T_c} = a_{ic} > 0 \ ; \ \frac{\partial f_i}{\partial T_o} = a_{io} > 0.$$

On the other hand, the function $x \longmapsto max(0, x)$ is only continuously differentiable on $\mathbb{R}\backslash\{0\}$ but it is Lipschitz continuous with a Lipschitz constant 1. We thus need to use Proposition 1.6 to prove that (5.12) is cooperative with respect to the control input u. Let $u, u' \in \mathbb{R}^4$ such that $u \geq u'$ (using the componentwise inequality), which means that for all $i \in \{1, 2, 3, 4\}$, $V_i \leq V_i'$. Since we only consider the cooling case ($T_u < T_i$) due to the experimental setup, we obtain for all $T \in \mathbb{R}^4$, $w \in \mathbb{R}^3$ and $\delta \in \{0, 1\}^8$:

$$f_i(T, u, w, \delta) - f_i(T, u', w, \delta) = \\ c_i(T_u - T_i)\left(\max\left(0, \frac{V_i - V_i^*}{\overline{V} - V_i^*}\right) - \max\left(0, \frac{V_i' - V_i^*}{\overline{V} - V_i^*}\right)\right) \geq 0.$$

Note that if we had chosen $V = [V_1, V_2, V_3, V_4]$ as the control input instead of $u = -V$, we would have obtained:

$$V_i \geq V_i' \Longrightarrow f_i(T, V, w, \delta) \leq f_i(T, V', w, \delta).$$

In this case, the global model (5.12) is not cooperative but it is still monotone with a partial ordering for the control input induced by the negative orthant $(\mathbb{R}^-)^4$: $\succeq_V \equiv \leq$.

Similarly to u, we can use Proposition 1.9 from Section 1.1.3 for the discrete disturbance δ. We consider $\succeq_\delta$ as the restriction to $\{0, 1\}^8$ of the componentwise inequality relation $\geq$ from $\mathbb{R}^8$. Since the heat source temperature is naturally assumed to be greater than the room temperature ($T_{s_i} > T_i$), we have:

$$\delta_{s_i} \geq \delta'_{s_i} \Longrightarrow \delta_{s_i} b_i(T_{s_i}^4 - T_i^4) \geq \delta'_{s_i} b_i(T_{s_i}^4 - T_i^4).$$

For the state of a door $\delta_{d_{ij}} \geq \delta'_{d_{ij}}$ with $T_i \geq T_j$, we have $f_i(T, u, w, \delta) = f_i(T, u, w, \delta')$ and $f_j(T, u, w, \delta) \geq f_j(T, u, w, \delta')$. We obtain symmetrical results for $T_i \leq T_j$. The model (5.12) is thus cooperative with respect to all discrete disturbances. Note that this could alternatively be proven by embedding the set $\{0, 1\}^8$ into the continuous space $\mathbb{R}^8$ and then apply Proposition 1.7 using the partial derivative defined on the continuous space.

Finally, we also guarantee that all inputs are bounded. It is clearly the case for the discrete disturbance δ which takes values in a finite set. For the identification of the parameters of the model in Section 5.2.3, we restricted the voltage of the underfloor fans to a range from 0 to $\overline{V} = 6\,\mathrm{V}$. We thus have $u \in [-\overline{V}, 0]^4$. For the continuous disturbance w, only the underfloor temperature T_u is controlled, though it may vary around its setpoint. In what follows, we simply assume that all three exogenous temperatures are bounded based on observation of the current conditions or forecast. These bounds define the robustness that we want to realize in our control strategies. Naturally, the correct-by-construction controller synthesis methods are only valid if w stays in its bounds. With these considerations, Assumption 1 is satisfied (and therefore Assumption $1'$ as well).

5.3.2 Contraction analysis

In this section, we study the notion of contracting systems [LS98], known under various other names such as convergent dynamics [PPvdWN04], extreme stability [Yos66] or incremental stability [Ang02]. A contracting system is defined as a

system whose trajectories from two initial states converge exponentially toward each other if the same input functions are applied. Using the results of [LS98] in the simple case where no variable change is operated, a contracting system is characterized as a system with a uniformly negative definite Jacobian. Writing this condition using the matrix measure induced by the infinity norm, it clearly appears that the Jacobian is uniformly negative definite if it is diagonally dominant with strictly negative diagonal elements [RdBS11]. We thus compute the diagonal elements of the Jacobian matrix:

$$\frac{\partial f_i}{\partial T_i}(T,u,w,\delta) = -\sum_{j\in\mathcal{N}_i^*} a_{ij} - 4\delta_{s_i} b_i T_i^3 - c_i \max\left(0, \frac{V_i - V_i^*}{\overline{V} - V_i^*}\right) - \sum_{j\in\mathcal{N}_i} \delta_{d_{ij}} d_{ij} \frac{3}{2} \max(0, T_j - T_i)^{1/2},$$

which is strictly negative. Then, the sum of all elements of a row of the Jacobian gives:

$$\sum_{j=1}^{4} \frac{\partial f_i}{\partial T_j} = -\sum_{j\in\mathcal{N}_i^*\setminus\mathcal{N}_i} a_{ij} - 4\delta_{s_i} b_i T_i^3 - c_i \max\left(0, \frac{V_i - V_i^*}{\overline{V} - V_i^*}\right) < 0.$$

The model (5.12) is thus contracting. Given the input functions $\mathbf{u}$, $\mathbf{w}$, $\boldsymbol{\delta}$, this means that the trajectories from two initial states T^a and T^b converge toward each other:

$$\lim_{t\to\infty} (\Phi(t, T^a, \mathbf{u}, \mathbf{w}, \boldsymbol{\delta}) - \Phi(t, T^b, \mathbf{u}, \mathbf{w}, \boldsymbol{\delta})) = 0.$$

Also, with constant inputs the system is autonomous and all its trajectories converge to a unique equilibrium [LS98]. Therefore, our contracting system has a static input-state characteristic as in Assumption 3.

5.4 Robust controlled invariance

Experimental conditions In this section, we apply to the experimental UFAD building the control strategies from Chapter 2 based on the notion of robust controlled invariant interval. Let us first give the operating conditions of the experiment presented in this section. As in the identification procedure in Section 5.2.3, the underfloor temperature T_u is regulated at 17 °C using a PID controller. To take into consideration possible variations from this setpoint due to the warmer air coming from the ceiling plenum, we assume $T_u \in [17, 18]$. For the bounds of the remaining components of w, both the outside temperature T_o and the ceiling temperature T_c are considered to vary in $[22, 25]$. In addition, $\delta \in [\underline{\delta}, \overline{\delta}]$ with $\underline{\delta} = \{0\}^8$ and $\overline{\delta} = \{1\}^8$ and $u \in [\underline{u}, \overline{u}]$ with $\underline{u} = \{-\overline{V}\}^4$ and $\overline{u} = \{0\}^4$. In these conditions, the minimal robust invariant interval $[\underline{T_0}, \overline{T_0}]$ from Theorem 2.3 is given by:

$$\overline{T_0} = k_T(\overline{u}, \overline{w}, \overline{\delta}) = \begin{pmatrix} 46.7 & 53.1 & 62.3 & 50.3 \end{pmatrix},$$
$$\underline{T_0} = k_T(\underline{u}, \underline{w}, \underline{\delta}) = \begin{pmatrix} 18.3 & 18.5 & 17.9 & 18.7 \end{pmatrix}.$$

The robust controlled invariance from Theorem 2.5 writes as follows:

$$\begin{cases} f(\overline{T}, \underline{u}, \overline{w}, \overline{\delta}) \leq 0, \\ f(\underline{T}, \overline{u}, \underline{w}, \underline{\delta}) \geq 0, \end{cases}$$

where $\leq$ and $\geq$ represent the componentwise inequalities on $\mathbb{R}^4$. Since $\overline{u}$ and $\underline{\delta}$ have all their components equal to 0, the second condition simply writes as

$$\frac{dT_i}{dt} = \sum_{j \in \mathcal{N}_i^*} a_{ij}(\underline{T}_j - \underline{T}_i),$$

but the first one involves all the non-linear terms of (5.11). In addition, unlike the simple $2D$ examples provided in Chapter 2, here the system is of dimension 4 and a visualization of the robust controlled invariance conditions in the state space is difficult to obtain. On the other hand, we can easily compute the lower bound of the set $\mathcal{UB} = \{T \in \mathbb{R}^4 \mid f(T, \underline{u}, \overline{w}, \overline{\delta}) \leq 0\}$ of allowed upper bounds for a robust controlled invariant interval. Similarly we can obtain the upper bound of the set $\mathcal{LB} = \{T \in \mathbb{R}^4 \mid f(T, \overline{u}, \underline{w}, \underline{\delta}) \geq 0\}$ of allowed lower bounds for a robust controlled invariant interval. This is done using the static input-state characteristic k_T:

$$\underline{\mathcal{UB}} = k_T(\underline{u}, \overline{w}, \overline{\delta}) = \begin{pmatrix} 22.6 & 22.7 & 25.6 & 24.4 \end{pmatrix},$$
$$\overline{\mathcal{LB}} = k_T(\overline{u}, \underline{w}, \underline{\delta}) = \begin{pmatrix} 21.6 & 21.6 & 21.7 & 21.8 \end{pmatrix}.$$

Since $\overline{\mathcal{LB}} \leq \underline{\mathcal{UB}}$, we know that $\mathcal{LB} \cap \mathcal{UB} = \emptyset$ which means that in these conditions, there exists no robustly locally stabilizable state as in Theorem 2.8. As we know from Section 5.2.4 that our model (5.11) is not perfect, we also try to obtain experimental values $\overline{\mathcal{LB}}^*$ and $\underline{\mathcal{UB}}^*$ from the building. For $\overline{\mathcal{LB}}$ we close all doors, turn off all fans and lamps and wait for an equilibrium. Similarly, $\overline{\mathcal{UB}}$ is obtained with all doors open, and all fans and lamps turned on. We obviously cannot set the extremal values of the exogenous temperatures w, which means that the obtained values are less restrictive than the worst-case that should be considered:

$$\overline{\mathcal{LB}}^* = \begin{pmatrix} 22.7 & 23.0 & 22.8 & 22.5 \end{pmatrix}, \qquad \underline{\mathcal{UB}}^* = \begin{pmatrix} 22.6 & 25.3 & 25.6 & 26.3 \end{pmatrix}.$$

We can thus choose a target interval $[\underline{T_f}, \overline{T_f}]$ containing the more restrictive values:

$$\underline{T_f} = \begin{pmatrix} 21 & 21 & 21 & 21 \end{pmatrix} \leq \overline{\mathcal{LB}}, \qquad \overline{T_f} = \begin{pmatrix} 24 & 26 & 26 & 27 \end{pmatrix} \geq \overline{\mathcal{UB}}^*,$$

and verify that it does satisfy the robust controlled invariance conditions from Theorem 2.5.

For the robust set stabilization, we consider the linear support functions described in Section 2.5.2 between the minimal robust invariant interval $[\underline{T_0}, \overline{T_0}]$ and the robust controlled invariant interval $[\underline{T_f}, \overline{T_f}]$ chosen above:

$$\begin{cases} \overline{X}(\lambda) = \lambda \overline{T_0} + (1 - \lambda)\overline{T_f}, \\ \underline{X}(\lambda) = \lambda \underline{T_0} + (1 - \lambda)\underline{T_f}. \end{cases}$$

Then, we verify numerically that $\underline{X}([0,1]) \subseteq \mathcal{LB}$ and $\overline{X}([0,1]) \subseteq \mathcal{UB}$. We use the stabilizing controller (2.10) from Theorem 2.11:

$$\mathsf{u}_i(T) = \underline{u_i} + (\overline{u_i} - \underline{u_i}) \frac{\overline{X}_i(\overline{\lambda}(T)) - T_i}{\overline{X}_i(\overline{\lambda}(T)) - \underline{X}_i(\underline{\lambda}(T))},$$

with $\underline{\lambda}(T) = \min\{\lambda \in [0,1] \mid \underline{X}(\lambda) \leq T\}$ and $\overline{\lambda}(T) = \min\{\lambda \in [0,1] \mid \overline{X}(\lambda) \geq T\}$. Since $\overline{u_i} = 0$ and $\underline{u_i} = -\overline{V}$, it can be simplified as follows:

$$\mathsf{u}_i(T) = \overline{V} \frac{\underline{X}_i(\underline{\lambda}(T)) - T_i}{\overline{X}_i(\overline{\lambda}(T)) - \underline{X}_i(\underline{\lambda}(T))}. \tag{5.13}$$

Note that this controller is equivalent to the decentralized linear controller (2.4) once the state T reaches the target interval $[\underline{T_f}, \overline{T_f}]$.

Control implementation We consider the following switching scenario for the lamps and doors:

$t = 0\,\text{min}$,	lamps 2 and 3 on;
$t = 3\,\text{min}$,	doors $1-2$ and $2-3$ open;
$t = 6\,\text{min}$,	lamp 4 on, door $3-4$ open;
$t = 12\,\text{min}$,	lamp 3 off, doors $2-3$ and $3-4$ closed;
$t = 18\,\text{min}$,	all lamps off, all doors closed;
$t = 34\,\text{min}$,	all lamps on, all doors open.

Figure 5.7 gives the corresponding experimental results of the UFAD building with the feedback controller (5.13). The left axis corresponds to the room temperatures T_i (blue) and the stabilization intervals $\underline{X}_i(\underline{\lambda}(T))$ and $\overline{X}_i(\overline{\lambda}(T))$ (red), measured in Celsius degrees and the right axis refers to the fan voltage $V_i = -u_i$ (green). Similarly to Figure 5.5, the vertical lines represent the switching instants of lamps and doors, using plain lines when the switching is occurring in the room.

We can first notice that the lower bound of the stabilization intervals is always equal to the lower bound of the target interval ($\underline{X}_i(\underline{\lambda}(T)) = \underline{T_f}$) since all temperatures start above the interval. The robust set stabilization is achieved during the first 6 minutes. This topic is discussed in more details on the next example, but we can note that as expected, all components of the upper bound of the stabilization interval $\overline{X}_i(\overline{\lambda}(T))$ are strictly decreasing until the state reaches the target interval.

After the stabilization, we can see that the feedback controller maintains the state in its prescribed bounds for all conditions of the disturbance δ covered by the switching scenario. In particular, between minutes 18 and 34 we have $\delta = \underline{\delta}$ and between minute 34 until the end, $\delta = \overline{\delta}$. In both of these extremal cases of the discrete disturbance, the state stays in the interval. Note that we cannot ensure the extremal conditions of all the disturbances since we have no control of w.

Robust set stabilization We present another experiment where the robust set stabilization lasts for 65 minutes and thus can be analyzed in more details. The reason for having a longer stabilization in Figure 5.8 is that the upper bound of

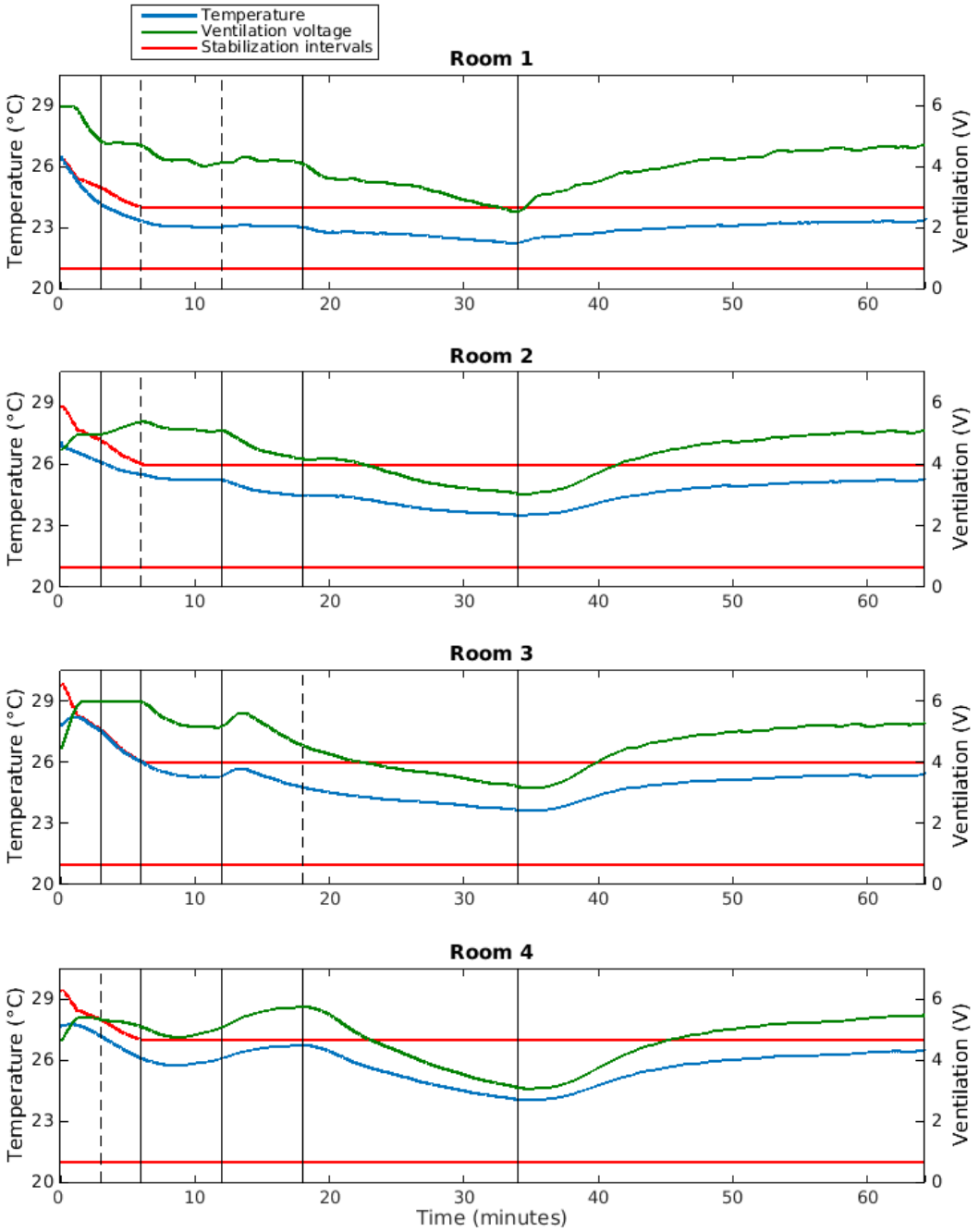


Figure 5.7 – Robust set stabilization and robust controlled invariance. Left axis: room temperature (blue) and stabilization intervals (red); right axis: fan voltage (green); vertical lines: switching times.

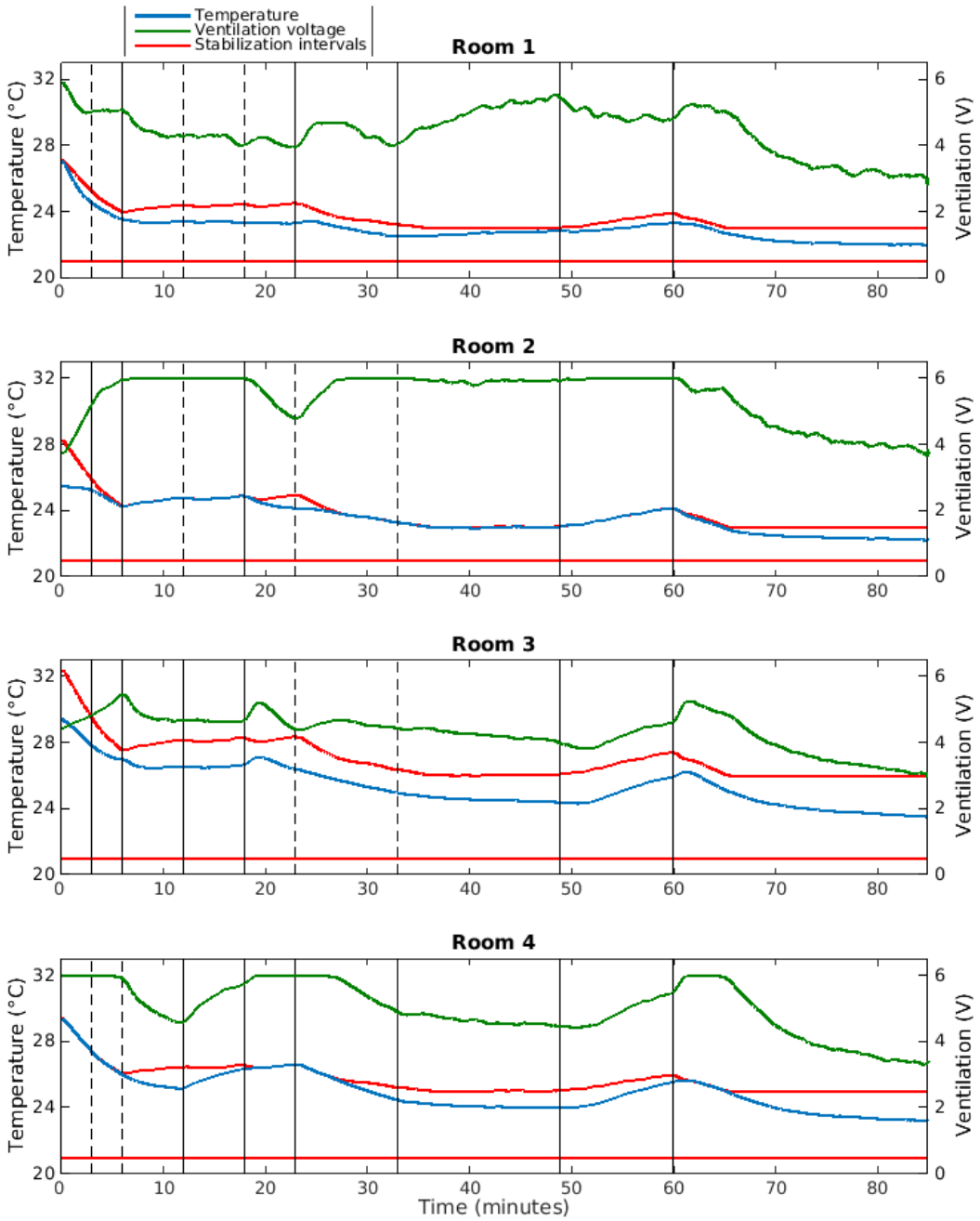


Figure 5.8 – Robust set stabilization for a non-robust controlled invariant interval (stabilization interval not strictly decreasing).

the stabilization interval is sometime increasing, while it is supposed to be strictly decreasing at all time. This comes from the chosen interval that is not sufficiently robust with respect to unmodeled dynamics: since the ventilation is always active with the controller (5.13), turning on a lamp creates a larger heat transfer than the modeled one (particularly in rooms 2 and 4). As this example corresponds to a case where the control is not achieved correctly, we do not detail the experimental conditions, but simply take advantage of this longer stabilization to discuss how it works.

At all time, the robust set stabilization corresponds to the robust controlled invariance in an interval $[\underline{X}(\underline{\lambda}(T)), \overline{X}(\overline{\lambda}(T))]$ with the current state T on its boundary. In the case of Figure 5.8, this means that at all time (before reaching the interval), there exists a room where $T_i = \overline{X}_i(\overline{\lambda}(T))$. For this room, the controller (5.13) applies the maximal ventilation to make sure that this temperature decreases, thus allowing a decrease of the upper bound of the stabilization interval. Therefore, during the stabilization, we always have a state component limiting the decrease of the upper bound of the stabilizing interval and for which the control input takes its extremal value. This can be seen in Figure 5.8 for rooms 2 and 4 alternatively during the whole stabilization process.

5.5 Symbolic control

In this section, we apply the two symbolic control methods from Chapters 3 and 4 to the experimental building: first the centralized method where a single symbolic abstraction is created to represent the whole system, then the compositional approach where one subsystem is created for the control of each underfloor fan.

5.5.1 Centralized approach

As stated in Section 5.4, it is difficult for this 4-dimensional model to obtain a visualization of the sets $\mathcal{LB}$ and $\mathcal{UB}$ defining the acceptable lower and upper bounds of robust controlled invariant intervals. Therefore, it is even more complicated to find an interval that may lead to a safe set strictly included in the interval: $\emptyset \neq Z_a^X \subsetneq [\underline{T}, \overline{T})$. In addition, looking for such intervals would not be wise: as we saw in Section 5.4, the unmodeled dynamics may cause the violation of the specifications even when the control strategy is theoretically correct. For these reasons, we consider the interval $[\underline{T}, \overline{T})$ such that

$$\underline{T} = \begin{pmatrix} 20 & 20 & 20 & 20 \end{pmatrix}, \qquad \overline{T} = \begin{pmatrix} 24 & 24 & 26 & 26 \end{pmatrix},$$

which is robust controlled invariant as in Theorem 2.5 when the exogenous temperatures have the following bounds: $T_c, T_o \in [21, 24]$ and $T_u \in [17, 18]$.

This interval is partitioned into $\alpha_x = 10$ intervals per dimension and the control set $[-\overline{V}, 0]^4$ is discretized into $\alpha_u = 4$ values per dimension. We thus obtain a symbolic abstraction with 10000 symbols and 256 control inputs. The sampling period is chosen as $\tau = 34\,\mathrm{s}$. This value is taken such that in the conditions of the fastest dynamics of the system (in our case, when the disturbance and ventilation

are at their maximum), the over-approximation of the reachable set from a symbol intersects some symbols that are not its immediate neighbors. As intended from the choice of the interval, the safety synthesis gives a safe set equal to the whole interval: $Z_a^X = [\underline{T}, \overline{T})$.

In addition to the safety specifications, we try to minimize a cost function that makes a tradeoff between three performance criteria:

$$g_a(s^k, u^k, u^{k-1}) = \frac{\|u^k\|}{\|\overline{u} - \underline{u}\|} + \frac{\|u^k - u^{k-1}\|}{\|\overline{u} - \underline{u}\|} + \frac{\|s_*^k - T_*\|}{\|(\overline{T} + \underline{T})/2\|}. \tag{5.14}$$

The first criterion aims to minimize the current value of the control input u^k. The second criterion considers the variations of the control input, which requires the use of an extended state containing the previous value of the control: $z^k = (s^k, u^{k-1})$ as in Remark 3.8. The third criterion takes the distance between the center s_*^k of the current symbol s^k and the center T_* of the interval $[\underline{T}, \overline{T}]$. To assign them equal weights, all three criteria are normalized with respect to the maximal value they can take as long as they satisfy the safety specification. Using these cost functions, the dynamic programming algorithm is run over a finite window of $N = 5$ sampling periods with a discount factor $\lambda = 0.5$ so that the constant part in the guaranteed upper bound in Theorem 3.10 is sufficiently small: $\frac{\lambda^{N+1}}{1-\lambda} \approx 3\%$.

The receding horizon control scheme is then applied to the result of the dynamic programming and we obtain a look-up table associating a control value $u \in \mathbb{R}^4$ to each value of the extended state (containing the current symbol and the previous control input). To control the experimental building, we thus only need to save the previous control input, measure the current temperature, convert it into the corresponding symbol and read the table. We run an experiment with the following switching scenario:

$t = 0\,\text{min}$, lamps off, doors closed;
$t = 5\,\text{min}$, lamp 1 on, doors $1-4$ and $3-4$ open;
$t = 20\,\text{min}$, lamp 2 on, door $2-3$ open;
$t = 35\,\text{min}$, lamps 3 and 4 on, door $1-2$ open;
$t = 50\,\text{min}$, lamps 2 and 4 off, door $3-4$ closed;
$t = 65\,\text{min}$, lamp 1 off, door $1-2$ closed.

As in the previous graphs, the blue curve in Figure 5.9 represents the measured temperature, the green curve is the fan voltage and the horizontal red lines are the lower and upper bounds of the target interval. We can see that the temperature in each room is correctly maintained between its bounds, except for room 1 where we can see some slight overshoots. These overshoots are explained by the unmodeled behaviors which are accumulating for a relatively long time due to the value of the sampling period $\tau = 34\,\text{s}$. Although the temperature is not always close to the center of the interval as it is affected by the disturbances, the other two performance criteria seem to be well satisfied: the ventilation is turned off when the state is far from the upper bound of the interval and its value is almost never changed twice in a row. In particular, it as been seen on other experiments that the performance criterion minimizing the variations of the control is the most important of the three. Indeed,

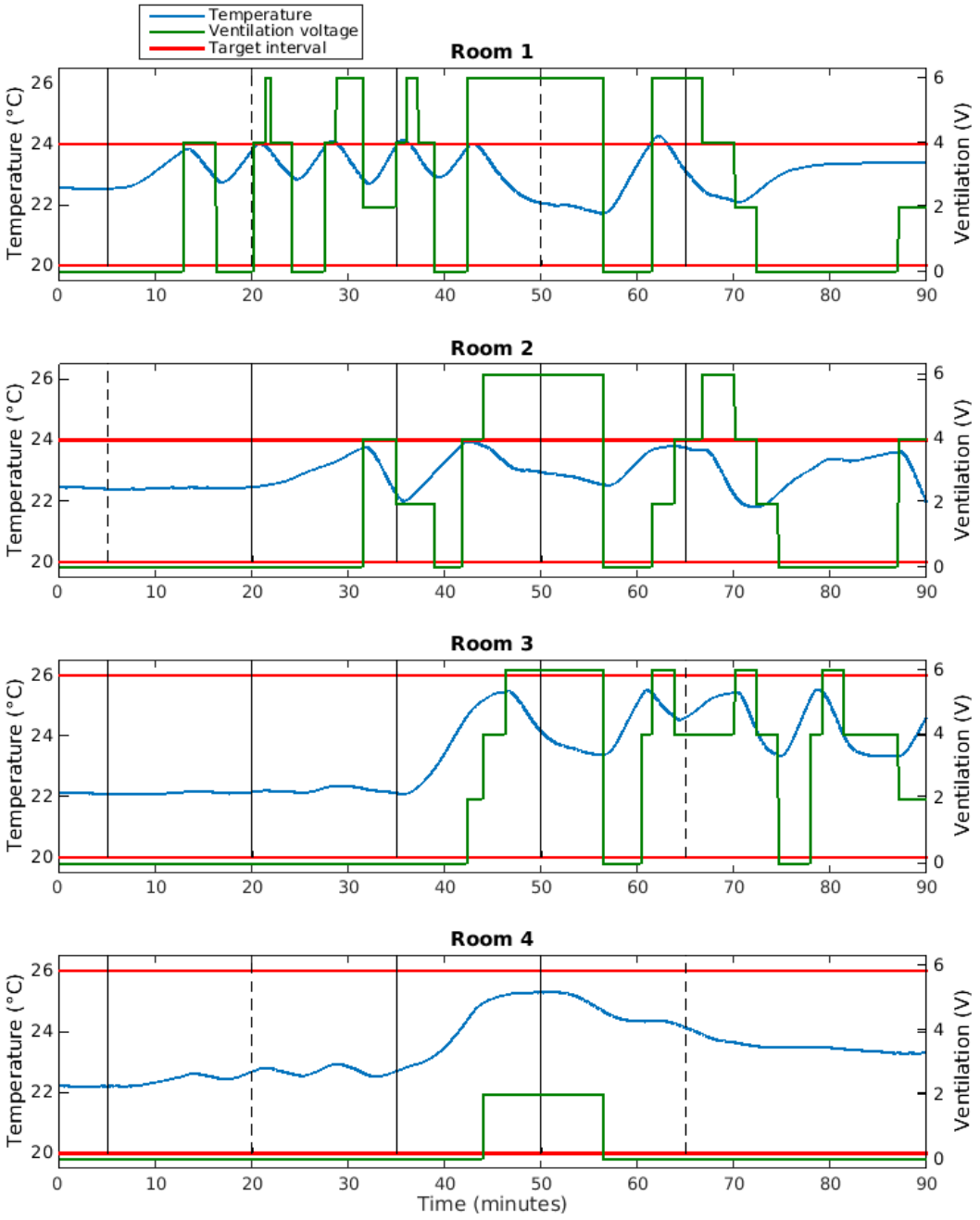


Figure 5.9 – UFAD experiment controlled with a centralized symbolic controller with 10^4 symbols and 4^4 control values. Control objective: minimize the control values, the control variations and the distance between the temperature and the center of the interval.

without it, the control input changes its value at each sampling time to minimize at best the other criteria, which may quickly damage the actuators.

With this first experimental implementation of a controller based on symbolic methods, we can immediately identify the main two challenges of such methods as discussed in Sections 3.1 and 4.1: robustness and scalability. Since this approach is based on the model of the system to control (including the model of the disturbances), even small unmodeled behaviors can render this correct-by-construction controller unreliable in an experimental case. We thus need to back it up with another controller (a simpler one, without performance guarantees) that can take over the feedback control when the symbolic method fails.

Regarding the complexity, even this simple example with 10 symbols and 4 control values per dimension on a system of 4 dimensions already reaches 2.56 million pairs symbol-input in the transition system S_a. The use of an extended state makes it even worse as in our case it duplicates all the iterations on the control input during the optimization. To give a general idea, the creation of the symbolic abstraction and the controller synthesis took a couple of days on a 3 GHz processor. This is highly problematic as the creation of this controller requires a prior knowledge or estimation of the range of the disturbances w, which means that the controller synthesized after 2 days may be useless if the estimation of w was inaccurate.

5.5.2 Compositional approach

To overcome the scalability and robustness problems identified in the previous section, we experiment the compositional methods presented in Chapter 4. This approach naturally solves the scalability issue by creating symbolic abstractions of systems with lower dimensions. As a consequence, a faster controller synthesis allows to consider a finer partition of the state space, thus requiring to choose a smaller value of the sampling period. The influence of the unmodeled behaviors can then be reduced by resetting their accumulation more often.

We consider the decomposition of the global model into four subsystems, each having a single state variable and the corresponding control input: for all $i \in \{1, 2, 3, 4\}$, $I_i = I_i^c = J_i = i$ and $I_i^o = \emptyset$. Similarly to the previous experiment, we estimate the bounds of the exogenous inputs w ($T_c, T_o \in [25, 28]$ and $T_u \in [17, 18]$) and choose a theoretically robust controlled invariant interval for these conditions:

$$\underline{T} = \begin{pmatrix} 23 & 23 & 23 & 23 \end{pmatrix}, \qquad \overline{T} = \begin{pmatrix} 26 & 26 & 28 & 28 \end{pmatrix}.$$

Since we are now considering four $1D$ subsystems instead of a single $4D$ model, we can safely consider higher precisions for the state partition and the control input discretization. Here we choose $\alpha_x = 20$ symbols and $\alpha_u = 9$ control values per dimension. Compared to the experiment with the centralized method in Section 5.5.1, the higher value of α_x implies the use of a lower sampling period $\tau = 10$ s, which should reduce the accumulation of unmodeled dynamics between two sampling times. For each subsystem, we consider a cost function similar to (5.14) but with an increased weight on the second performance criterion (minimization of control variations) to prevent the behavior seen in room 1 of Figure 5.9 where the temperature repeatedly exceeds the prescribed bound and the controller reduces the value of the ventilation

as soon at the temperature is brought back into the interval. In these conditions, the computation of all four abstractions and synthesis of the four corresponding controllers only takes 1.1 s. This experiment is run on a similar switching scenario:

$t = 0\,\text{min}$, lamp 1 on, doors $1-4$ and $3-4$ open;
$t = 15\,\text{min}$, lamp 2 on, door $2-3$ open;
$t = 30\,\text{min}$, lamps 3 and 4 on, door $1-2$ open;
$t = 45\,\text{min}$, lamps 2 and 4 off, door $3-4$ closed;
$t = 60\,\text{min}$, lamp 1 off, door $1-2$ closed;
$t = 75\,\text{min}$, lamp 3 off, doors $1-4$ and $2-3$ closed.

We can see in Figure 5.10 that the higher weight on the minimization of the control variations has a significant influence since there is no more than 4 changes of the ventilation per room over 100 min. All temperatures are maintained below the upper bound of the interval and in particular we avoid the oscillating behaviors observed in rooms 1 and 3 of the experiment in Figure 5.9 for the centralized method. However, we can see at the end of the experiment in rooms 2 and 3 that the temperature goes below the lower bound of the interval for a short time. This appears as one of the drawbacks of increasing the precision α_x of the partition: the symbolic model has a more accurate information on the current state and since it assumes a perfect model, it waits for lower temperatures before stopping the ventilation. Then, the presence of unmodeled delay induced by filtering the noise on the measured temperatures has a bigger influence on such model than on coarser models which need to be more conservative. On other experiments with finer partitions ($\alpha_x = 50$ with $\tau = 5\,\text{s}$ and $\alpha_x = 200$ with $\tau = 1\,\text{s}$, which can be easily computed in 3 and 10 s, respectively), this delay was observed to have a significant influence.

5.6 Concluding remarks

On this relatively small system with 4 states and 4 control inputs, we saw in Section 5.5.1 that for the centralized symbolic method, the computational cost is already too high to reach a sufficiently detailed model. On this point, the compositional approach provides a significant improvement since the controller synthesis can be achieved with high precision in less than a minute for the decomposition in $1D$ subsystems ($I_i = I_i^c = J_i = i$ and $I_i^o = \emptyset$). Although the results are not given in this chapter, we also considered another decomposition into $3D$ subsystems centered on a room $I_i^c = J_i = i$ but where we also observe the state of both neighbor rooms $I_i^o = \mathcal{N}_i$. This solution is thus a tradeoff between both previous methods, but it still involves a low computation time: e.g. with $\alpha_x = 10$ and $\alpha_u = 4$, the controller is synthesized in less than 6 s while the centralized approach needs more than 2 days. This low complexity of the compositional approach allows us to consider the synthesis of an automatic method similar to model predictive control, where we would synthesize a controller using tighter bounds on the disturbance w, apply it for a short period of time and repeat after measuring the new value of the disturbance.

We can also note that the results based on robust controlled invariance in Chapter 2 describe the possibility to control a system rather than provide an actual

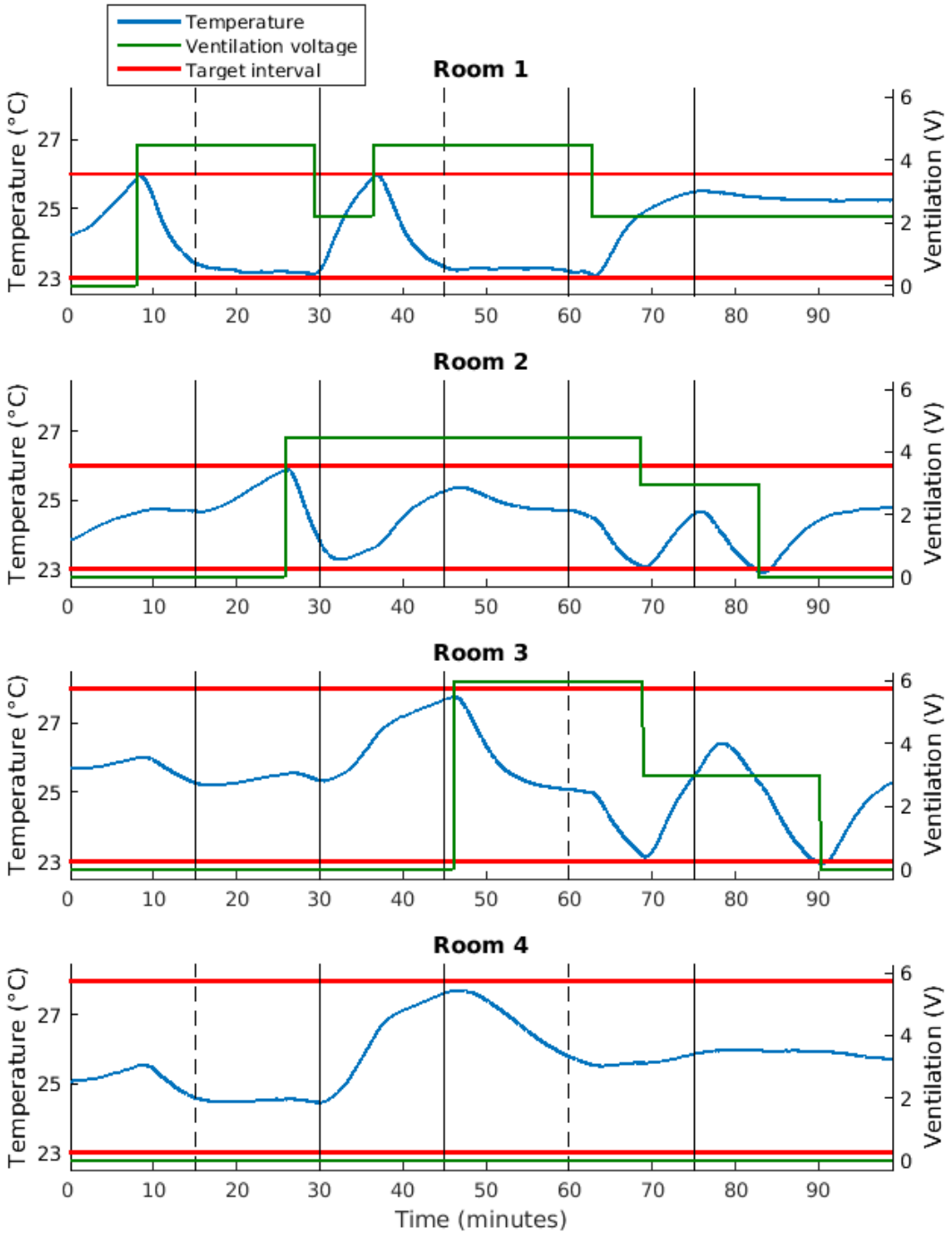


Figure 5.10 – UFAD experiment controlled with a compositional symbolic method using $1D$ subsystems, each with 20 symbols and 9 control values.

control strategy to maintain the state in an interval. Therefore, as it was done in this chapter and the illustrations of the previous chapters, this degree of freedom allows us to combine these results with the control strategies based on symbolic methods by facilitating the choice of the target interval. This is particularly useful when we apply compositional methods where we often cannot find partial safe sets $Z_c \subsetneq \mathcal{P}^0$ in a non-robust controlled invariant interval. In addition, once a robust controlled invariant interval is chosen, we can combine it with the robust set stabilization results from Section 2.5 to stabilize the state inside this interval before using the more efficient symbolic control.

Conclusion and perspectives

In this thesis, we focus on the robust control of a class of cooperative systems subject to disturbances. This problem is approached from two angles: firstly with the notion of robust controlled invariant interval for continuous-time systems, secondly using symbolic methods to synthesize a discrete controller on a finite abstraction of the system.

In Chapter 2, we consider the notion of robust controlled invariance, which describes the ability to maintain the state of the system in a set for any value of the disturbances.

With the additional local control assumption where each control input only has a direct influence on a single state variable, a robust controlled invariant interval can be characterized by using only the sign of the vector field of the system with the extremal values of all inputs. Two results are derived from this. The first one describes a robustly locally stabilizable state using arbitrarily small robust controlled invariant intervals containing this state. The second consequence concerns robust set stabilization, where we can stabilize the state of the system initialized outside a robust controlled invariant interval by using a decreasing family of robust controlled invariant intervals.

Although these results imply some constraints on the control values when the state is on the boundary of its target interval, they mainly provide information on the ability to control the system and leave a large degree of freedom on the choice of the actual control strategy.

In Chapter 3, we are interested in symbolic methods to synthesize a controller realizing a safety specification.

This method consists in creating a finite abstraction of the sampled dynamics before synthesizing a discrete controller realizing the specifications on the abstraction. Using the cooperativeness of the system, we prove that this controller also satisfies the specifications on the original system. We have shown that this method provides a safe set that is larger than the one that could be obtained with the robust controlled invariance in Chapter 2. Since several control strategies realize the safety specification, we run an optimization over the safe controls to minimize a cost function on a finite horizon and we apply a receding horizon control scheme. The obtained controller provides performance guarantees on the total accumulated cost of an infinite trajectory.

In Chapter 4, we introduce a compositional method to address the scalability

issue of the centralized symbolic approach from Chapter 3.

The dynamics are decomposed into subsystems that give a partial description of the global model where some of the states and inputs are not observed. In addition, some of the state components are only modeled to increase the precision of the subsystem but are not controlled. The symbolic abstraction and synthesis methods are applied to each subsystem under the assume-guarantee obligations that the safety specification is realized for the unobserved and the uncontrolled states. The composition of the obtained controllers are proven to realize the global safety specification and to provide similar performance guarantees to the ones of the centralized method. The safety and performance results are naturally weaker than the one from Chapter 3 due to the loss of information, but they are obtained with a significantly reduced complexity, thus widening the range of possible applications for these symbolic methods.

Throughout the thesis, all the results are illustrated with numerical simulations on two simple examples. In addition, in Chapter 5, we evaluate our methods on the temperature regulation in a small-scale experimental building equipped with UnderFloor Air Distribution.

On this application, we see that all our control methods can be combined: use the robust controlled invariance to choose an interval where the safety is guaranteed for the symbolic control and apply the robust set stabilization to bring the state inside this interval before using the symbolic controller. The significant complexity reduction of the compositional approach is also shown, since even on this 4-dimensional system, a satisfying precision is out of reach of the centralized symbolic approach, while the compositional method only needs a few seconds to obtain a very high precision.

The work presented in this thesis provides numerous directions for future development. We describe below those that we think are the most important or most promising ones.

Symbolic control In this thesis, we focused the control specifications solely on safety to provide a comparison between the symbolic methods and the robust controlled invariance. It would thus be interesting to look at the modifications of our current method required by the use of other specifications such as temporal logic formulas. We can note that this change does not only modify the controller synthesis but also the abstraction task which needs a prior knowledge of the control objective to choose what information of the state space can be abstracted.

Automatizing the choice of the sampling period based on the precision of the state space partition could significantly reduce the trial and error phase before obtaining satisfying results on the safe set. Since this choice significantly depends on the dynamics of the system, we shall look for a similar relation as the one given in [SP94] for viability kernels of autonomous systems, where the sampling and the partition parameters are linked through an inequality involving the Lipschitz constant and the supremum of the vector field of the system.

In the current version of the compositional method presented in Chapter 4, the state components modeled in a subsystem are either controlled or simply observed to

increase the model accuracy. However, for the control input, all modeled components are used for control. It would thus be interesting to see if we can obtain similar results when we also model some control inputs which are not used for control, similarly to what is done for the state. This problem may need the introduction of an additional assume-guarantee obligation stating that the observed but uncontrolled inputs do not take values that play against the safety specification of the current subsystem. Another approach is to consider *conditionally competitive* subsystems [CH07], where the uncontrolled inputs play against this subsystem but not at the risk of violating the specifications of their main subsystems.

Adaptive symbolic control With the scalability issue of the symbolic methods partially solved in the compositional approach, we could consider integrating it in a larger adaptive control framework similar to model predictive control, using two time scales. On the smaller scale, the symbolic controller is applied as previously. On the larger scale, we measure the current value of the disturbance and create a tight estimation of its bounds for the near future (small multiple of the large time step), then synthesize a new symbolic controller corresponding to these conditions to be applied until the next measure of the disturbance. This structure could address several problems currently limiting the use of our symbolic method and thus widen the range of its possible applications.

Robustness Although the symbolic controllers in this methods are not robust when taken separately, the global approach adapts the model used for synthesis to the current value of the disturbance. Thus it is robust to a wider range of disturbance than can be considered in a single symbolic controller without loosing the safety.

Precision This is the same reason as in the previous point: instead of taking a large estimation of the disturbance, we consider tighter bounds around the current value, resulting in smaller over-approximation of the reachable sets and therefore more degrees of freedom for the controller synthesis.

Cooperativeness Since we know that the synthesized controller are only used for a short duration, we could focus on local cooperative behaviors of a non-cooperative system (e.g. a ventilation system that can provide both warm and cold air is not globally cooperative since an increase of the ventilation can have a positive or negative effect on the temperature, but on a shorter period we know that the air cannot be both warm and cold).

This global controller may need to be combined with a robust set stabilization strategy if the current state is not in the safe set of the newly computed controller.

Applications In the UFAD application of Chapter 5, the first required improvement is on the model of the temperature variations since all our control methods are model-based and suffer from unmodeled dynamics. Several leads are given in Section 5.2.4, but we should keep in mind that the mathematical properties of the model (particularly the cooperativeness) need to be preserved.

Since the heat transfers linked to the heat sources and doors have a significant influence on the UFAD system, it would be interesting to add an estimator of the

current value of the discrete disturbances δ, particularly if these disturbances can be assumed to have a dwell time. With this information, we can apply the controller synthesized on the more accurate model corresponding to the current value of δ and thus obtain better performances. This estimator may require to be combined with a robust set stabilization controller to handle the transition between two conditions, or when the estimator provides a false result.

Other fields can be considered for an application of the control strategies developed in this thesis. Recently, we have been particularly interested in vehicle platooning: when heavy duty trucks drive in close proximity to reduce the air drag and therefore the fuel consumption. In our initial study of a possible model for such multi-vehicle system [AGJT14], it seems to satisfy the monotonicity property. In addition, such systems are particularly adapted to our compositional approach since each vehicle is only directly influenced by two others.

Bibliography

[AGJT14] Assad Alam, Ather Gattami, Karl H. Johansson, and Claire J. Tomlin. Guaranteeing safety for heavy duty vehicle platooning: Safe set computations and experimental evaluations. *Control Engineering Practice*, 24:33–41, 2014.

[AH99] Rajeev Alur and Thomas A. Henzinger. Reactive modules. *Formal Methods in System Design*, 15(1):7–48, 1999.

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

[ALA] Jacob Apkarian, Hervé Lacheray, and Amin Abdossalami. *Coupled Tanks Workbook, student version.* Quanser Inc. `http://www.quansershare.com/Home/Search?contentID=301`.

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

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

[ASVR99] Betzaida Arguello-Serrano and Miguel Velez-Reyes. Nonlinear control of a heating, ventilating, and air conditioning system with thermal load estimation. *IEEE Transactions on Control Systems Technology*, 7(1):56–63, 1999.

[ATS09] Alessandro Abate, Ashish Tiwari, and Shankar Sastry. Box invariance in biologically-inspired dynamical systems. *Automatica*, 45(7):1601–1610, 2009.

[Aub91] Jean-Pierre Aubin. *Viability theory.* Birkhäuser, 1991.

[BD03] Fred S. Bauman and Allan Daly. *Underfloor air distribution (UFAD) design guide.* ASHRAE, 2003.

[Ber95] Dimitri P. Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific Belmont, MA, 1995.

[BG13] Ismail Belgacem and Jean-Luc Gouzé. Global stability of enzymatic chains of full reversible Michaelis-Menten reactions. *Acta biotheoretica*, 61(3):425–436, 2013.

[Bla99] Franco Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.

[BM69] Giuseppe Basile and Giovanni Marro. Controlled and conditioned invariant subspaces in linear system theory. *Journal of Optimization Theory and Applications*, 3(5):306–315, 1969.

[BM07] Franco Blanchini and Stefano Miani. *Set-theoretic methods in control*. Springer, 2007.

[CA15] Samuel Coogan and Murat Arcak. Efficient finite abstraction of mixed monotone systems. In *Hybrid Systems: Computation and Control*, pages 58–67. Springer, 2015.

[CGG11] Javier Cámara, Antoine Girard, and Gregor Gössler. Synthesis of switching controllers using approximately bisimilar multiscale abstractions. In *Proceedings of the* 14^{th} *international conference on Hybrid systems: computation and control*, pages 191–200. ACM, 2011.

[CH07] Krishnendu Chatterjee and Thomas A. Henzinger. Assume-guarantee synthesis. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 261–275. Springer, 2007.

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

[CL08] Christos G. Cassandras and Stephane Lafortune. *Introduction to discrete event systems*. Springer Science & Business Media, 2008.

[DLAS05] Patrick De Leenheer, David Angeli, and Eduardo D. Sontag. On predator-prey systems and small-gain theorems. *Mathematical Biosciences and Engineering*, 2(1):25–42, 2005.

[DLB14] Xuchu Ding, Mircea Lazar, and Calin Belta. LTL receding horizon control for finite deterministic systems. *Automatica*, 50(2):399–408, 2014.

[dR98] Willem-Paul de Roever. The need for compositional proof systems: A survey. In *Compositionality: the significant difference*, pages 1–22. Springer, 1998.

[EED12] Directive 2012/27/EU of the European Parliament and of the Council of 25 october 2012 on energy efficiency. Official Journal of the European Union, 2012.

[EPB02] Directive 2002/91/EC of the European Parliament and of the Council of 16 december 2002 on the energy performance of buildings. Official Journal of the European Communities, 2002.

[EPB10] Directive 2010/31/EU of the European Parliament and of the Council of 19 may 2010 on the energy performance of buildings. Official Journal of the European Union, 2010.

[ESS06] German A. Enciso, Hal L. Smith, and Eduardo D. Sontag. Nonmonotone systems decomposable into monotone systems with negative feedback. *Journal of Differential Equations*, 224(1):205–227, 2006.

[Fre05] Goran Fedja Frehse. *Compositional verification of hybrid systems using simulation relations*. PhD thesis, Radboud University Nijmegen, 2005.

[GDV14] Reza Ghaemi and Domitilla Del Vecchio. Control for safety specifications of systems with imperfect information on a partial order. *IEEE Transactions on Automatic Control*, 59(4):982–995, April 2014.

[GH94] Jean-Luc Gouzé and Karl P. Hadeler. Monotone flows and order intervals. *Nonlinear World*, 1:23–34, 1994.

[Gir14] Antoine Girard. Approximately bisimilar abstractions of incrementally stable finite or infinite dimensional systems. In *Proceedings of the IEEE Conference on Decision and Control*, 2014.

[GLG08] Antoine Girard and Colas Le Guernic. Zonotope/hyperplane intersection for hybrid systems reachability analysis. In *Hybrid Systems: Computation and Control*, pages 215–228. Springer, 2008.

[GP07] Antoine Girard and George J. Pappas. Approximation metrics for discrete and continuous systems. *IEEE Transactions on Automatic Control*, 52(5):782–798, 2007.

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

[HL98] Maher Hamdi and Gerard Lachiver. A fuzzy control system based on the human sensation of thermal comfort. In *Proceedings of the Conference on Fuzzy Systems*, volume 1, pages 487–492, 1998.

[HQR98] Thomas A. Henzinger, Shaz Qadeer, and Sriram K. Rajamani. You assume, we guarantee: Methodology and case studies. In *Computer aided verification*, pages 440–451. Springer, 1998.

[HQRT98] Thomas A. Henzinger, Shaz Qadeer, Sriram K. Rajamani, and Serdar Taşiran. An assume-guarantee rule for checking simulation. In *Formal Methods in Computer-Aided Design*, pages 421–431. Springer, 1998.

[HS95] John M. House and Theodore F. Smith. Optimal control of building and HVAC systems. In *Proceedings of the American Control Conference*, volume 6, pages 4326–4330, 1995.

[HS05] Morris W. Hirsch and Hal L. Smith. Monotone dynamical systems. *Handbook of differential equations: ordinary differential equations*, 2:239–357, 2005.

[IEA14] Key world energy statistics. International Energy Agency, 2014.

[Jon83] Cliff B. Jones. Tentative steps toward a development method for interfering programs. *ACM Transactions on Programming Languages and Systems*, 5(4):596–619, 1983.

[Kam32] Erich Kamke. Zur Theorie der Systeme gewöhnlicher Differentialgleichungen. II. *Acta Mathematica*, 58(1):57–85, 1932.

[Kra68] Mark Aleksandrovich Krasnoselskii. *The operator of translation along the trajectories of differential equations*. American Mathematical Society, 1968.

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

[LDGR07] Michel De Lara, Luc Doyen, Thérèse Guilbaud, and Marie-Joëlle Rochet. Monotonicity properties for the viable control of discrete-time systems. *Systems & Control Letters*, 56(4):296–302, 2007.

[Lev02] Geoff J. Levermore. *Building Energy Management Systems: Applications to Low-Energy HVAC and Natural Ventilation Control*. Taylor & Francis, 2nd edition, 2002.

[LO14] Jun Liu and Necmiye Ozay. Abstraction, discretization, and robustness in temporal logic control of dynamical systems. In *Proceedings of the* 17^{th} *international conference on Hybrid systems: computation and control*, pages 293–302. ACM, 2014.

[LS98] Winfried Lohmiller and Jean-Jacques E. Slotine. On contraction analysis for nonlinear systems. *Automatica*, 34(6):683–696, 1998.

[LSL61] Joseph La Salle and Solomon Lefschetz. *Stability by Liapunov's direct methods*. Elsevier, 1961.

[Lun94] Jan Lunze. Qualitative modelling of linear dynamical systems with quantized state measurements. *Automatica*, 30(3):417–431, 1994.

[MC81] Jayadev Misra and K. Mani Chandy. Proofs of networks of processes. *IEEE Transactions on Software Engineering*, (4):417–426, 1981.

[MGWa] Pierre-Jean Meyer, Antoine Girard, and Emmanuel Witrant. Robust controlled invariance for monotone systems: application to ventilation regulation in buildings. Provisionally accepted in Automatica.

[MGWb] Pierre-Jean Meyer, Antoine Girard, and Emmanuel Witrant. Safety control with performance guarantees of cooperative systems using compositional abstractions. Submitted at the 5^{th} IFAC Conference on Analysis and Design of Hybrid Systems.

[MGW13] Pierre-Jean Meyer, Antoine Girard, and Emmanuel Witrant. Controllability and invariance of monotone systems for robust ventilation automation in buildings. In *Proceedings of the* 52^{nd} *IEEE Conference on Decision and Control*, pages 1289–1294, 2013.

[MGW15] Pierre-Jean Meyer, Antoine Girard, and Emmanuel Witrant. Poster: Symbolic control of monotone systems, application to ventilation regulation in buildings. In *Proceedings of the* 18^{th} *ACM International Conference on Hybrid Systems: Computation and Control*, pages 281–282, 2015.

[MNGW13] Pierre-Jean Meyer, Hosein Nazarpour, Antoine Girard, and Emmanuel Witrant. Poster abstract: Robust controlled invariance for UFAD regulation. In *Proceedings of the* 5^{th} *ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, pages 1–2, 2013.

[MNGW14] Pierre-Jean Meyer, Hosein Nazarpour, Antoine Girard, and Emmanuel Witrant. Experimental implementation of UFAD regulation based on robust controlled invariance. In *Proceedings of the* 13^{th} *European Control Conference*, pages 1468–1473, 2014.

[MPS10] Faye C. McQuiston, Jerald D. Parker, and Jeffrey D. Spitler. *Heating, ventilating, and air conditioning: analysis and design.* Wiley, 6th edition, 2010.

[MR02] Thomas Moor and Jörg Raisch. Abstraction based supervisory controller synthesis for high order monotone continuous systems. In *Modelling, Analysis, and Design of Hybrid Systems*, pages 247–265. Springer, 2002.

[MSGT08] Hossein Mirinejad, Seyed Hossein Sadati, Maryam Ghasemian, and Hamid Torab. Control techniques in heating, ventilating and air conditioning systems. *Journal of Computer Science*, 4(9):777–783, 2008.

[MT00] Ian Mitchell and Claire J. Tomlin. Level set methods for computation in hybrid systems. In *Hybrid Systems: Computation and Control*, pages 310–323. Springer, 2000.

[Mül27] Max Müller. Über das Fundamentaltheorem in der Theorie der gewöhnlichen Differentialgleichungen. *Mathematische Zeitschrift*, 26(1):619–645, 1927.

[OPJ+10] Frauke Oldewurtel, Alessandra Parisio, Colin N. Jones, Manfred Morari, Dimitrios Gyalistras, Markus Gwerder, Vanessa Stauch, Beat Lehmann, and Katharina Wirth. Energy efficient building climate control using stochastic model predictive control and weather predictions. In *Proceedings of the American Control Conference*, pages 5100–5105, 2010.

[PGT08] Giordano Pola, Antoine Girard, and Paulo Tabuada. Approximately bisimilar symbolic models for nonlinear control systems. *Automatica*, 44(10):2508–2516, 2008.

[PLOP08] Luis Perez-Lombard, Jose Ortiz, and Christine Pout. A review on buildings energy consumption information. *Energy and buildings*, 40(3):394–398, 2008.

[Pnu77] Amir Pnueli. The temporal logic of programs. In 18^{th} *Annual Symposium on Foundations of Computer Science*, pages 46–57, 1977.

[PPS06] Nir Piterman, Amir Pnueli, and Yaniv Sa'ar. Synthesis of reactive (1) designs. In *Verification, Model Checking, and Abstract Interpretation*, pages 364–380. Springer, 2006.

[PPvdWN04] Alexey Pavlov, Alexander Pogromsky, Nathan van de Wouw, and Henk Nijmeijer. Convergent dynamics, a tribute to Boris Pavlovich Demidovich. *Systems & Control Letters*, 52(3):257–261, 2004.

[RdBS11] Giovanni Russo, Mario di Bernardo, and Jean-Jacques E. Slotine. A graphical approach to prove contraction of nonlinear circuits and systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 58(2):336–348, 2011.

[Rei09] Gunther Reißig. Computation of discrete abstractions of arbitrary memory span for nonlinear sampled systems. In *Hybrid Systems: Computation and Control*, pages 306–320. Springer, 2009.

[Rei10] Gunther Reißig. Abstraction based solution of complex attainability problems for decomposable continuous plants. In 49^{th} *IEEE Conference on Decision and Control*, pages 5911–5917, 2010.

[RM09] James Blake Rawlings and David Q. Mayne. *Model predictive control: Theory and design*. Nob Hill Pub., 2009.

[RMC09] Nacim Ramdani, Nacim Meslem, and Yves Candau. A hybrid bounding method for computing an over-approximation for the reachable set of uncertain nonlinear systems. *IEEE Transactions on Automatic Control*, 54(10):2352–2364, 2009.

[RMC10] Nacim Ramdani, Nacim Meslem, and Yves Candau. Computing reachable sets for uncertain nonlinear monotone systems. *Nonlinear Analysis: Hybrid Systems*, 4(2):263–278, 2010.

[RT] Matthias Rungger and Paulo Tabuada. A notion of robustness for cyber-physical systems. arXiv preprint arXiv:1310.5199.

[RW87] Peter J. Ramadge and W. Murray Wonham. Supervisory control of a class of discrete event processes. *SIAM journal on control and optimization*, 25(1):206–230, 1987.

[SK03] Olaf Stursberg and Bruce H. Krogh. Efficient representation and computation of reachable sets for hybrid systems. In *Hybrid Systems: Computation and Control*, pages 482–497. Springer, 2003.

[Smi88] Hal L. Smith. Systems of ordinary differential equations which generate an order preserving flow. a survey of results. *SIAM review*, 30(1):87–113, 1988.

[Smi95] Hal L. Smith. *Monotone dynamical systems: an introduction to the theory of competitive and cooperative systems*, volume 41. American Mathematical Soc., 1995.

[Smi06] Hal L. Smith. The discrete dynamics of monotonically decomposable maps. *Journal of Mathematical Biology*, 53(4):747–758, 2006.

[Son07] Eduardo D. Sontag. Monotone and near-monotone biochemical networks. *Systems and Synthetic Biology*, 1(2):59–87, 2007.

[Son08] Eduardo D. Sontag. Input to state stability: Basic concepts and results. In *Nonlinear and optimal control theory*, pages 163–220. Springer, 2008.

[SP94] Patrick Saint-Pierre. Approximation of the viability kernel. *Applied Mathematics and Optimization*, 29(2):187–209, 1994.

[SP05] Sigurd Skogestad and Ian Postlethwaite. *Multivariable feedback control: analysis and design*. Wiley, 2^{nd} edition, 2005.

[Tab09] Paulo Tabuada. *Verification and control of hybrid systems: a symbolic approach*. Springer, 2009.

[TC98] Jason Teeter and Mo-Yuen Chow. Application of functional link neural network to hvac thermal dynamic system identification. *IEEE Transactions on Industrial Electronics*, 45(1):170–176, 1998.

[TCRM14] Paulo Tabuada, Sina Yamac Caliskan, Matthias Rungger, and Rupak Majumdar. Towards robustness for cyber-physical systems. *Automatic Control, IEEE Transactions on*, 59(12):3151–3163, 2014.

[TH10] Marija Trčka and Jan L. M. Hensen. Overview of HVAC system simulation. *Automation in Construction*, 19(2):93–99, 2010.

[TI08] Yuichi Tazaki and Jun-ichi Imura. Bisimilar finite abstractions of interconnected systems. In *Hybrid Systems: Computation and Control*, pages 514–527. Springer, 2008.

[TPL04] Paulo Tabuada, George J. Pappas, and Pedro Lima. Compositional abstractions of hybrid control systems. *Discrete event dynamic systems*, 14(2):203–238, 2004.

[TSH01] Harry L. Trentelman, Anton A. Stoorvogel, and Malo Hautus. *Control theory for linear systems*. Springer Verlag, 2001.

[vdMAB+92] Koos van der Maas, Francis Allard, Dominique Bienfait, Fariborz Haghighat, Georges Liébecq, Roger Pelletret, Luk Vandaele, and Richard Walker. *Air flow through large openings in buildings*. Laboratoire d'énergie solaire et de physique du bâtiment, 1992.

[VV01] Mahesh Viswanathan and Ramesh Viswanathan. Foundations for circular compositional reasoning. In *Automata, Languages and Programming*, pages 835–847. Springer, 2001.

[WDMPB10] Emmanuel Witrant, Piergiuseppe Di Marco, Pangun Park, and Corentin Briat. Limitations and performances of robust control over WSN: UFAD control in intelligent buildings. *IMA Journal of Mathematical Control and Information*, 27(4):527–543, 2010.

[WHZB99] Qing-Guo Wang, Chang-Chieh Hang, Yong Zhang, and Qiang Bi. Multivariable controller auto-tuning with its application in hvac systems. In *Proceedings of the American Control Conference*, volume 6, pages 4353–4357, 1999.

[WLW05] Johnny K. W. Wong, Heng Li, and S. W. Wang. Intelligent building research: a review. *Automation in Construction*, 14(1):143–159, 2005.

[WM70] W. Murray Wonham and A. Stephen Morse. Decoupling and pole assignment in linear multivariable systems: a geometric approach. *SIAM Journal on Control*, 8(1):1–18, 1970.

[Yos66] Taro Yoshizawa. *Stability theory by Liapunov's second method*. Mathematical Society of Japan, 1966.

Abstract — This thesis provides new control strategies that deal with the heterogeneous and nonlinear dynamics describing the temperature regulation in buildings to obtain a tradeoff between comfort and energy efficiency. We thus focus on the robust control of cooperative systems with bounded disturbances. We first solve this problem with the notion of robust controlled invariant interval, which describes a set where the state can be maintained for any value of the disturbances. A second approach provides dedicated symbolic methods to synthesize a discrete controller on a finite abstraction of the system, realizing safety specifications combined with a performance optimization. We first present a centralized symbolic method using the system dynamics provided by the physical model. To address its limitation in terms of scalability, a compositional approach is considered, where the symbolic abstraction and synthesis methods are applied to partial descriptions of the system under the assume-guarantee obligation that the safety specification is realized for all uncontrolled states. In the final part, the proposed controllers are combined and evaluated on the temperature regulation for an experimental building equipped with UnderFloor Air Distribution.

Keywords: cooperative system, robust controlled invariance, abstraction-based synthesis, compositional synthesis, intelligent building.

Résumé — Cette thèse fournit de nouvelles stratégies de contrôle pouvant s'attaquer aux phénomènes hétérogènes et non-linéaires qui décrivent la régulation de la température dans les bâtiments afin d'obtenir un compromis entre le confort et l'efficacité énergétique. Nous nous intéressons donc au contrôle robuste de systèmes coopératifs avec perturbations bornées. Nous résolvons d'abord ce problème grâce à la notion d'intervalle invariant contrôlé robuste, décrivant un ensemble dans lequel l'état peut être maintenu quelle que soit la valeur des perturbations. Une seconde approche décrit des méthodes symboliques pour la synthèse d'un contrôleur discret sur une abstraction finie du système, réalisant une spécification de sûreté associée à l'optimisation des performances. Nous présentons d'abord une méthode symbolique centralisée utilisant les dynamiques du système correspondant au modèle physique. Pour résoudre ses limitations en termes de passage à l'échelle, nous considérons une approche compositionnelle où les méthodes symboliques d'abstraction et de synthèse sont appliquées à des descriptions partielles du système, sous des obligations de type *assume-guarantee* supposant que la sûreté est satisfaite pour tous les états non-contrôlés. Dans la dernière partie, les contrôleurs présentés sont combinés et évalués dans le cadre d'une régulation de température pour un bâtiment expérimental équipé de la solution *UnderFloor Air Distribution*.

Mots clés : système coopératif, invariance contrôlée robuste, synthèse à base d'abstraction, synthèse compositionnelle, bâtiment intelligent.

Laboratoire Jean Kuntzmann
51 rue des Mathématiques, BP 53
38041 Grenoble cedex 09

GIPSA-lab
11 rue des Mathématiques, BP 46
38402 Saint Martin d'Hères cedex