



TECHNIQUES
DE L'INGÉNIEUR

Réf. : **S7467 V1**

Date de publication :
10 septembre 2024

Approches symboliques pour le contrôle des systèmes non linéaires

Cet article est issu de : **Automatique - Robotique | Automatique et ingénierie système**

par **Antoine GIRARD, Pierre-Jean MEYER,**
Adnane SAOUD

Mots-clés

Contrôle des systèmes non linéaires | méthodes formelles | modèle symbolique | synthèse discrète

Résumé Cet article traite de la synthèse de contrôleurs pour des systèmes non linéaires soumis à des contraintes sur les états et la commande et à des perturbations bornées, et pour des spécifications telles que la sûreté, l'atteignabilité ou des propriétés plus complexes formulées à l'aide d'automates ou de logiques temporelles. Dans ce contexte, les approches symboliques, qui reposent sur l'abstraction du système par un modèle symbolique (avec un nombre fini d'états et de commandes), permettent la synthèse automatique de contrôleurs certifiés « corrects par construction ». Cet article expose de manière didactique les éléments clés de ces approches (abstraction, synthèse et concrétisation des contrôleurs) et présente une synthèse des thématiques avancées de ce domaine de recherche dynamique.

Keywords

Control of nonlinear systems | formal methods | symbolic model | discrete synthesis

Abstract This article deals with controller synthesis for nonlinear systems subject to constraints on states and control, as well as bounded disturbances, and for specifications such as safety, reachability, or more complex properties formulated using automata or temporal logics. In this context, symbolic approaches, based on the abstraction of the system through a symbolic model (with a finite number of states and controls), enable the automatic synthesis of controllers certified as "correct by construction." This article presents in a didactic manner the key elements of these approaches (abstraction, synthesis, and concretization of controllers) and provides an overview of advanced themes in this dynamically evolving research field.

Pour toute question :

Service Relation clientèle
Techniques de l'Ingénieur
Immeuble Pleyad 1
39, boulevard Ornano
93288 Saint-Denis Cedex

Par mail :
infos.clients@teching.com

Par téléphone :
00 33 (0)1 53 35 20 20

Document téléchargé le : **25/09/2024**

Pour le compte : **7200106152 - éditions ti // les PARCOURS PRATIQUES // 2.59.188.28**

© Techniques de l'Ingénieur | Tous droits réservés

Approches symboliques pour le contrôle des systèmes non linéaires

par **Antoine GIRARD**

Directeur de recherche CNRS
Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes,
91190, Gif-sur-Yvette, France

Pierre-Jean MEYER

Chargé de recherche du développement durable
Univ Gustave Eiffel, COSYS-ESTAS, F-59666 Villeneuve d'Ascq, France

et **Adnane SAOUD**

Professeur Assistant
College of Computing, Mohammed VI Polytechnic University (UM6P), Benguerir, Maroc

| | |
|--|---------------------|
| 1. Abstraction symbolique d'un système complexe | S 7 467 - 3 |
| 1.1 Méthodologie d'abstraction symbolique | — 3 |
| 1.2 Contrôleur symbolique et concrétisation..... | — 6 |
| 2. Synthèse d'un contrôleur symbolique | — 7 |
| 2.1 Propriétés de sûreté..... | — 7 |
| 2.2 Autres spécifications..... | — 9 |
| 3. Exemple numérique : contrôle d'un robot mobile | — 11 |
| 3.1 Modèle et spécification..... | — 11 |
| 3.2 Abstraction et synthèse de contrôleur | — 11 |
| 3.3 Simulations numériques | — 12 |
| 4. Thèmes avancés en commande symbolique | — 13 |
| 4.1 Abstractions efficaces et compositionnelles | — 13 |
| 4.2 Abstractions complètes | — 14 |
| 4.3 Abstractions basées sur les données..... | — 14 |
| 4.4 Applications du contrôle symbolique | — 15 |
| 5. Conclusion..... | — 16 |
| 6. Glossaire | — 16 |
| 7. Sigles, notations et symboles..... | — 17 |
| Pour en savoir plus | Doc. S 7 467 |

Dans le domaine de l'automatique, l'utilisation de modèles mathématiques les plus fidèles possibles (non linéaires, avec contraintes sur les états et commandes, avec perturbations, etc.) permet généralement d'augmenter la fiabilité et la confiance en les résultats obtenus suite à l'analyse de ces modèles. Malheureusement, cette complexité des modèles rend l'étape de synthèse de contrôleurs d'autant plus difficile, voire parfois impossible. D'autre part, bien qu'il existe une importante bibliographie considérant des objectifs de contrôle classiques de l'automatique continue tels que la stabilité, ces méthodes ne sont pas les plus adaptées pour traiter d'autres types de spécifications telles que la sûreté, l'atteignabilité, ou la riche gamme de propriétés pouvant être décrites par des automates ou des formules de logique temporelle que l'on retrouve dans les domaines de l'informatique et de la robotique.

Les approches symboliques présentées dans cet article ont pour but de s'attaquer à l'intersection de ces deux points. Ces approches s'appuient sur trois étapes successives :

- l'abstraction du système dynamique continu en un modèle symbolique discret décrit par un système dynamique avec un nombre fini d'états et de commandes ;
- la synthèse d'un contrôleur dans le domaine discret ;
- la concrétisation du contrôleur symbolique pour son application sur le système continu de départ.

Contrairement aux méthodes de commande prédictive, l'intégralité de ces trois étapes est réalisée hors ligne, et l'application en ligne du contrôleur concrétisé se fait par le biais d'un simple tableau de correspondance.

Ces approches à base d'abstraction symbolique ont de nombreux avantages. Le premier est qu'elles peuvent traiter une très large gamme de systèmes dynamiques non linéaires complexes (avec perturbations, incertitudes, contraintes d'état ou de commande) d'une manière complètement automatique puisque la synthèse de contrôleur dans le domaine discret est indépendante de la forme et de la complexité des dynamiques continues. Elles permettent également de s'intéresser à une nouvelle gamme de spécifications plus variées et plus complexes que celles traditionnellement considérées en automatique continue, telles que des formules de logique temporelle. D'autre part, les algorithmes utilisés pour la synthèse dans le domaine discret sont issus des domaines des méthodes formelles et du model checking, et permettent d'obtenir des garanties formelles de la satisfaction des spécifications même après la concrétisation du contrôleur vers le domaine continu ; le contrôleur est ainsi certifié « correct par construction ». Enfin, puisque la première étape d'abstraction symbolique s'appuie sur des méthodes à base de surapproximations de l'ensemble des comportements réels du système, l'approche globale se retrouve dotée d'une certaine robustesse intrinsèque en supplément de la prise en compte des perturbations et incertitudes du modèle. En revanche, la principale limitation des approches symboliques est leur complexité exponentielle en la dimension de l'espace d'état, ce qui les rend difficiles à appliquer sur des systèmes de grande dimension. Mais de nombreux travaux, présentés dans la section 4 de cet article, ont déjà été réalisés pour réduire cette complexité et permettre un meilleur passage à l'échelle des méthodes symboliques.

Bien qu'il existe de nombreuses variations et améliorations possibles sur la manière de mettre en œuvre les trois étapes constituant les approches symboliques, l'objectif de cet article est de fournir une présentation synthétique et pédagogique de la méthode au cœur de toutes ces approches, afin d'être facilement utilisable. Nous commençons par l'introduction des définitions nécessaires à la description des étapes 1 et 3 de l'approche symbolique, c'est-à-dire l'abstraction du problème du domaine continu vers le domaine discret, et la concrétisation du contrôleur discret vers le domaine continu. Nous présentons ensuite l'étape 2 pour la synthèse d'un contrôleur symbolique dans le domaine discret, pour plusieurs types d'objectifs de contrôle, en mettant l'accent sur les spécifications de sûreté. L'approche complète est alors illustrée par un exemple numérique sur un véhicule modélisé comme un monocycle devant visiter plusieurs régions (avec contraintes sur leur ordre) et en éviter d'autres. La présentation de cet article se focalisant sur la méthode de base des approches symboliques, dans la section 4 nous proposons une ouverture sur les nombreuses variations et améliorations des méthodes symboliques ayant été récemment publiées dans la littérature scientifique, en donnant une attention particulière aux méthodes permettant de réduire ou de s'abstraire des problèmes de complexités.

1. Abstraction symbolique d'un système complexe

Dans cette section, nous présentons une méthodologie générale pour le calcul d'un modèle symbolique d'un système dynamique (étape 1 d'*Abstraction* dans la figure 1). Nous présentons ensuite le théorème fondamental du contrôle symbolique qui certifie que les contrôleurs synthétisés sur la base du modèle symbolique sont « corrects par construction » pour le système original (étape 3 de *Concrétisation* dans la figure 1). L'approche symbolique présentée dans cette section résulte d'une synthèse de nombreux travaux présents dans la littérature. Les personnes souhaitant approfondir les aspects mathématiques présentés dans cette section pourront commencer par les travaux [1] et [2].

Dans cet article, nous considérerons un système dynamique non linéaire de la forme suivante :

$$x(t+1) = f(x(t), u(t), w(t)), \quad x(t) \in \mathbb{X}, \quad u(t) \in \mathbb{U}, \quad w(t) \in \mathbb{W} \quad (1)$$

où $x(t)$, $u(t)$, $w(t)$ dénotent l'état, la commande et la perturbation du système à l'instant $t \in \mathbb{N}$. Les ensembles $\mathbb{X} \subseteq \mathbb{R}^{n_x}$, $\mathbb{U} \subseteq \mathbb{R}^{n_u}$, $\mathbb{W} \subseteq \mathbb{R}^{n_w}$, sont supposés compacts (c'est-à-dire fermés et bornés) et représentent les contraintes sur l'état, la commande et la perturbation du système. Dans cet article, nous ne considérerons que des systèmes dynamiques en temps discret. Toutefois, l'approche symbolique peut aussi être utilisée pour des systèmes en temps continu, généralement via une discrétisation temporelle.

Un **modèle symbolique** est une « abstraction finie » de la dynamique de (1). Plus précisément, il s'agit d'une approximation de (1) par un système dynamique dont les ensembles d'états et de commandes sont des ensembles finis, c'est-à-dire qu'il n'existe qu'un nombre fini de valeurs possibles pour l'état et la commande du modèle symbolique. Ainsi, un modèle symbolique peut s'écrire sous la forme suivante :

$$\xi(t+1) \in g(\xi(t), \sigma(t)), \quad \xi(t) \in \Xi, \quad \sigma(t) \in \Sigma \quad (2)$$

où $\xi(t)$ et $\sigma(t)$ dénotent l'état et la commande du modèle symbolique à l'instant $t \in \mathbb{N}$, Ξ et Σ sont des ensembles finis. Notons

que le modèle symbolique n'inclut en général pas de terme de perturbations dont l'influence peut être capturée par le caractère multivalué de la fonction de transition g . En effet, pour un état ξ et une commande symbolique σ , $g(\xi, \sigma)$ est un sous-ensemble de Ξ représentant l'ensemble des états successeurs potentiels.

1.1 Méthodologie d'abstraction symbolique

Nous décrivons ici une approche pour calculer un modèle symbolique du système décrit dans (1). L'**abstraction du système** se fait en deux étapes :

- 1/ définition des ensembles d'états et de commandes symboliques Ξ et Σ par une discrétisation des ensembles d'états et de commandes de (1) ;
- 2/ définition de la fonction de transition multivaluée g par surapproximation de la dynamique de (1).

1.1.1 Discrétisation de l'état et de la commande

Considérons une partition de l'ensemble d'états \mathbb{X} : soit $\mathbb{X}_1, \dots, \mathbb{X}_m$ une collection de sous-ensembles de \mathbb{X} vérifiant :

$$\mathbb{X} = \mathbb{X}_1 \cup \dots \cup \mathbb{X}_m \quad \text{et} \quad \mathbb{X}_i \cap \mathbb{X}_j = \emptyset, \quad \forall i \neq j.$$

De plus, on note $\mathbb{X}_0 = \mathbb{R}^{n_x} \setminus \mathbb{X}$. On définit alors l'ensemble des états symboliques comme $\Xi = \{0, \dots, m\}$ où l'état symbolique $\xi \in \Xi$ représente l'ensemble des états $x \in \mathbb{X}_\xi$. On associe également l'**interface d'abstraction** $q : \mathbb{R}^{n_x} \rightarrow \Xi$ qui associe à tout $x \in \mathbb{R}^{n_x}$ l'état symbolique lui correspondant :

$$\forall x \in \mathbb{R}^{n_x}, \quad \forall \xi \in \Xi, \quad (q(x) = \xi \Leftrightarrow x \in \mathbb{X}_\xi).$$

Pour la définition de l'ensemble de commandes symboliques, on considère un nombre fini de commandes $u_1, \dots, u_l \in \mathbb{U}$. On définit alors l'ensemble des commandes symboliques comme $\Sigma = \{1, \dots, l\}$ où la commande symbolique $\sigma \in \Sigma$ représente la commande $u_\sigma \in \mathbb{U}$. On définit ainsi l'**interface de concrétisation** $p : \Sigma \rightarrow \mathbb{U}$ telle que pour tout $\sigma \in \Sigma$, $p(\sigma) = u_\sigma$.

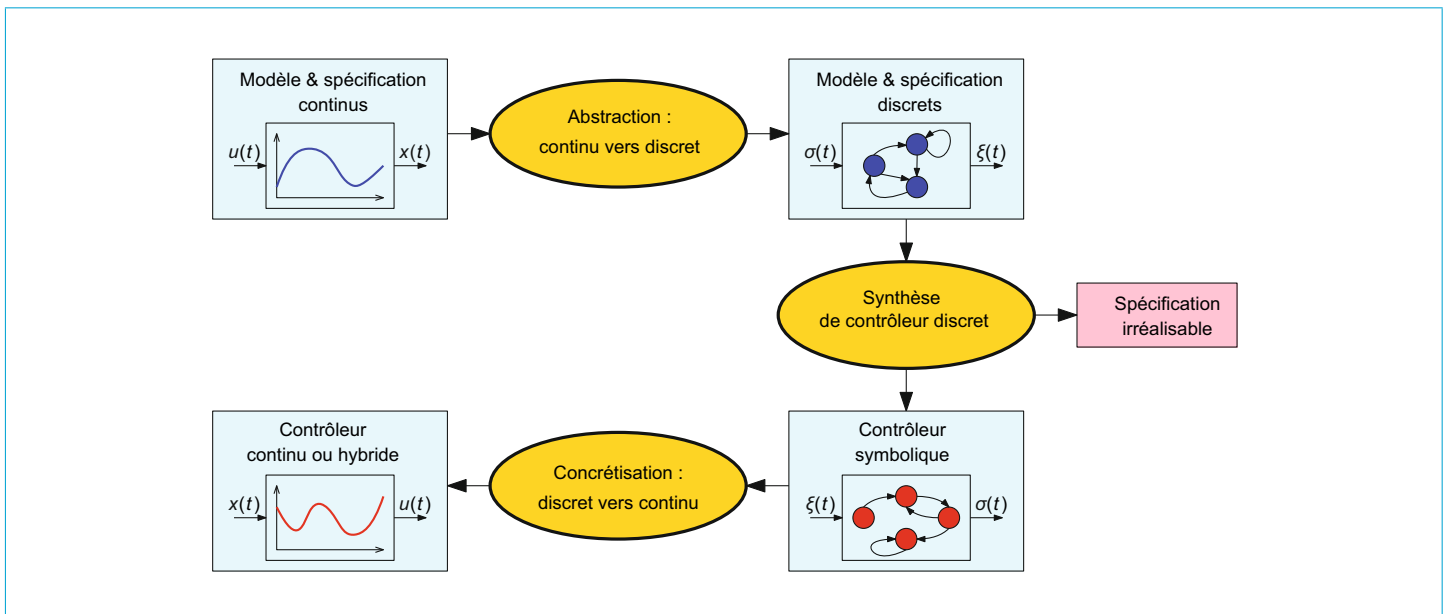


Figure 1 – Aperçu de l'approche symbolique pour le contrôle

1.1.2 Approximation de la dynamique

Les ensembles d'états et de commandes symboliques Ξ et Σ étant maintenant proprement définis, il nous reste à spécifier la fonction de transition g . Pour un état et une commande symbolique $\xi \in \Xi$, $\sigma \in \Sigma$, $g(\xi, \sigma)$ dénote l'ensemble des états symboliques vers lesquels le modèle symbolique peut évoluer.

Pour que le modèle symbolique soit utilisable pour synthétiser des contrôleurs pour le système décrit dans (1), il faut que la dynamique symbolique capture l'ensemble des comportements possibles de (1). Pour ce faire, il est nécessaire de calculer pour chaque $\xi \in \Xi$, $\sigma \in \Sigma$ un ensemble $\mathbb{Y}_{\xi, \sigma} \subseteq \mathbb{R}^{n_x}$ vérifiant l'inclusion ci-dessous :

$$\mathbb{Y}_{\xi, \sigma} \supseteq f(\text{cl}(\mathbb{X}_\xi), u_\sigma, \mathbb{W}) := \{f(x, u_\sigma, w) \mid x \in \text{cl}(\mathbb{X}_\xi), w \in \mathbb{W}\} \quad (3)$$

où $\text{cl}(\mathbb{X}_\xi)$ dénote l'adhérence de l'ensemble \mathbb{X}_ξ . Ainsi, $\mathbb{Y}_{\xi, \sigma}$ est un ensemble contenant l'ensemble des états vers lesquels (1) peut évoluer depuis un état $x \in \text{cl}(\mathbb{X}_\xi)$, sous l'action de la commande u_σ et l'effet d'une perturbation $w \in \mathbb{W}$. Le calcul des ensembles $\mathbb{Y}_{\xi, \sigma}$ représente le principal point de difficulté dans le calcul du modèle symbolique ; plusieurs méthodes ont été développées à cet effet (section 1.1.3).

Ensuite, la dynamique du modèle symbolique peut être définie de la manière suivante :

$$g(\xi, \sigma) = \{\xi^+ \in \Xi \mid \mathbb{Y}_{\xi, \sigma} \cap \text{cl}(\mathbb{X}_{\xi^+}) \neq \emptyset\}.$$

La procédure d'approximation de la dynamique de (1) est illustrée dans la figure 2.

Remarque 1

Les choix des discrétisations de l'état et de la commande sont importants pour le succès de l'approche symbolique. En effet, des discrétisations trop grossières (*i.e.* avec m et l petits) résulteront en un modèle symbolique très imprécis qui ne permettra pas de synthétiser un contrôleur. À l'opposé, des discrétisations trop fines (*i.e.* avec m et l grands) résulteront en un modèle symbolique de très grande complexité qui ne pourra pas être traité en temps raisonnable. Notons qu'il existe des méthodes de raffinement itératif des discrétisations afin de trouver le bon compromis précision/complexité (section 4.1.1). Nous décrivons ci-dessous un critère simple pour déterminer si une discrétisation est trop grossière. Supposons que (1) satisfait la condition de mobilité suivante :

$$\forall x \in \mathbb{X}, \exists u \in \mathbb{U}, \forall w \in \mathbb{W}, x \neq f(x, u, w). \quad (4)$$

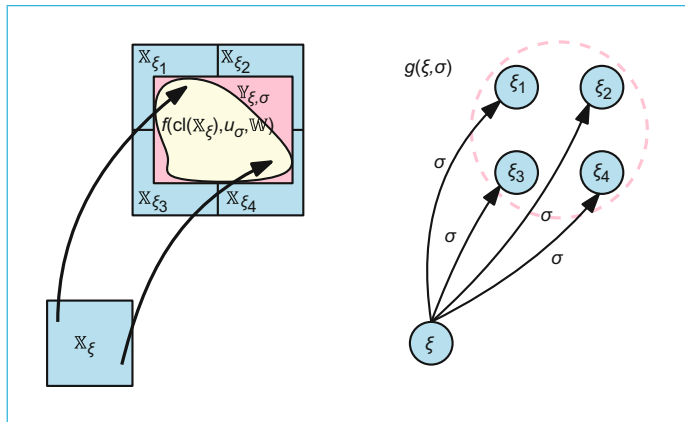


Figure 2 – Calcul de la dynamique symbolique par approximation de la dynamique de (1)

Remarque 1 (Suite)

La condition ci-dessus signifie qu'il est toujours possible de choisir une commande de (1) pour modifier la valeur courante de l'état du système, une condition nécessaire pour satisfaire des propriétés autres que celles de sûreté. Pour pouvoir utiliser le modèle symbolique pour ce type de propriétés, il est nécessaire qu'il satisfasse également la condition de mobilité :

$$\forall \xi \in \Xi \setminus \{0\}, \exists \sigma \in \Sigma, \xi \notin g(\xi, \sigma). \quad (5)$$

Si les discrétisations de l'état et de la commande sont trop grossières, il est possible que (5) ne soit pas vérifiée même si (4) l'est. En revanche, si (4) est vérifiée, la condition (5) le sera également pour toute discrétisation suffisamment fine.

Remarque 2

Le modèle symbolique construit suivant la procédure décrite dans cette section est en fait valable pour n'importe quel système de la forme (1) où f est remplacée par une fonction \tilde{f} vérifiant pour chaque $\xi \in \Xi$, $\sigma \in \Sigma$

$$\{\xi^+ \in \Xi \mid \tilde{f}(\text{cl}(\mathbb{X}_\xi), u_\sigma, \mathbb{W}) \cap \text{cl}(\mathbb{X}_{\xi^+}) \neq \emptyset\} \subseteq g(\xi, \sigma).$$

Ainsi, il apparaît qu'un modèle symbolique ne représente pas uniquement une abstraction d'un seul système dynamique mais d'un ensemble de systèmes dynamiques. Les contrôleurs synthétisés par les approches symboliques disposent alors par construction de certaines marges de robustesse.

Un exemple simple : le thermostat

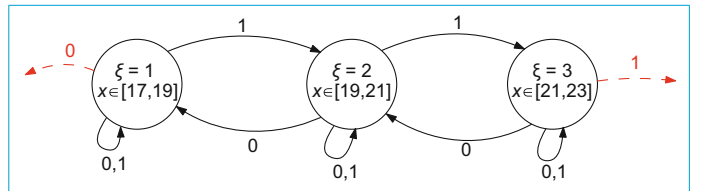
Considérons un modèle de l'évolution de température dans une pièce décrit par :

$$x(t+1) = x(t) + a(w(t) - x(t)) + u(t)b(T - x(t)),$$

où $x \in \mathbb{R}$ est la température moyenne de la pièce, $u \in \{0,1\}$ est la commande du thermostat (arrêt/marche), $w \in [10,15]$ est la température extérieure, $T = 30$ est la température du chauffage quand il est en marche, et $a = 0,1$ et $b = 0,2$ sont les constantes thermiques respectivement associées à la perte de température vers l'extérieur et au chauffage lorsqu'il est allumé.

Notre objectif est d'appliquer la méthode d'abstraction symbolique présentée dans cette section pour maintenir la température x dans l'intervalle $\mathbb{X} = [17,23]$ malgré la perturbation $w \in [10,15]$. L'intervalle \mathbb{X} est partitionné en trois sous-intervalles de même taille $\mathbb{X}_1 = [17,19]$, $\mathbb{X}_2 = [19,21]$, $\mathbb{X}_3 = [21,23]$ représentés par les états symboliques $\xi = 1, 2, 3$. L'état symbolique $\xi = 0$ représente alors toutes les autres températures que l'on souhaite éviter $\mathbb{X}_0 = \mathbb{R} \setminus \mathbb{X}$. L'ensemble des commandes $\mathbb{U} = \{0,1\}$ est déjà un ensemble discret que nous conservons ainsi par simplicité en dénotant $\Sigma = \mathbb{U}$.

La dynamique du système étant très simple (linéaire et de dimension 1), nous sommes dans un des rares cas où les ensembles atteignables du système peuvent être évalués de manière exacte, sans avoir recours aux méthodes de surapproximation de la section 1.1.3. La fonction de transition g du modèle symbolique obtenu après l'application de la section 1.1.2 peut ainsi être décrite par le graphe ci-dessous (les transitions rouges sont celles vers l'état symbolique $\xi = 0$ non représenté, que l'on souhaite éviter).



La synthèse d'un contrôleur de sûreté pour maintenir la température dans $\mathbb{X} = [17, 23]$ (décrite plus en détails dans la section 2.1) consiste alors à interdire la commande $\sigma = 0$ depuis l'état symbolique $\xi = 1$ et la commande $\sigma = 1$ depuis $\xi = 3$.

1.1.3 Analyse d'atteignabilité

Le problème décrit par (3) d'évaluer un ensemble $\mathbb{Y}_{\xi, \sigma}$ contenant l'ensemble $f(\text{cl}(\mathbb{X}_{\xi}), u_{\sigma}, \mathbb{W})$ atteignable par le système (1) appartient au domaine de l'**analyse d'atteignabilité** [3], et plus spécifiquement aux méthodes de surapproximation. Une des méthodes les plus simples et les plus répandues au sein des approches à base d'abstraction symbolique s'appuie sur une propriété de monotonie [4] du système. Nous présentons ici l'intuition et l'utilisation de cette méthode dans le cas particulier d'un système monotone dit **coopératif**, c'est-à-dire que f est différentiable et chaque élément de ses dérivées partielles $\frac{\partial f}{\partial x}(x, u, w)$ et $\frac{\partial f}{\partial w}(x, u, w)$ est positif.

Dans ce cas, le système satisfait la propriété suivante pour tout $x, \bar{x} \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$ et $w, \bar{w} \in \mathbb{R}^{n_w}$:

$$x \geq \bar{x}, w \geq \bar{w} \Rightarrow f(x, u, w) \geq f(\bar{x}, u, \bar{w}),$$

où les inégalités sont appliquées à chaque dimension des vecteurs comparés. En supposant que les ensembles $\text{cl}(\mathbb{X}_{\xi})$ et \mathbb{W} sont définis par des intervalles multidimensionnels (c'est-à-dire qu'il existe $\underline{x}, \bar{x}, \underline{w}, \bar{w}$ tels que $\text{cl}(\mathbb{X}_{\xi}) = [\underline{x}, \bar{x}] \subseteq \mathbb{R}^{n_x}$, $\mathbb{W} = [\underline{w}, \bar{w}] \subseteq \mathbb{R}^{n_w}$), il nous suffit d'appliquer à deux reprises l'implication ci-dessus pour obtenir un nouvel intervalle surapproximant l'ensemble des successeurs atteignables par le système (1) :

$$f(\text{cl}(\mathbb{X}_{\xi}), u_{\sigma}, \mathbb{W}) \subseteq [f(\underline{x}, u_{\sigma}, \underline{w}), f(\bar{x}, u_{\sigma}, \bar{w})].$$

Ainsi, comme illustré dans la figure 3a, il est suffisant de ne calculer que deux successeurs de (1) pour obtenir le plus petit intervalle (en termes d'inclusion) contenant intégralement l'ensemble atteignable.

Malgré ces avantages, la principale limitation de cette approche est qu'elle n'est applicable qu'aux systèmes satisfaisant la propriété de monotonie. Ci-après, nous présentons une méthode

alternative couvrant une plus large classe de systèmes tout en restant simple d'utilisation. Cette approche correspond à une version en temps discret de la méthode basée sur les bornes de croissance dans [2], et est applicable à tout système dont f est différentiable et ses dérivées sont uniformément bornées : il existe $D_x \in \mathbb{R}^{n_x \times n_x}$ et $D_w \in \mathbb{R}^{n_x \times n_w}$ telles que pour tout $x \in \text{cl}(\mathbb{X}_{\xi})$, $u \in \mathbb{U}$, $w \in \mathbb{W}$, $\left| \frac{\partial f}{\partial x}(x, u, w) \right| \leq D_x$ et $\left| \frac{\partial f}{\partial w}(x, u, w) \right| \leq D_w$, où les inégalités sur les matrices sont interprétées coefficient par coefficient.

Pour un intervalle d'états initiaux $\text{cl}(\mathbb{X}_{\xi}) = [\underline{x}, \bar{x}]$, on dénote par $x^* = \frac{\underline{x} + \bar{x}}{2}$ son centre, et par $\delta_x = \frac{\bar{x} - \underline{x}}{2}$ le vecteur de sa demi-largeur. On définit similairement w^* et δ_w pour un intervalle de perturbations $\mathbb{W} = [\underline{w}, \bar{w}]$. Ainsi, pour un système aux dérivées bornées comme décrit ci-dessus, on peut surapproximer l'ensemble atteignable par un intervalle :

$$f(\text{cl}(\mathbb{X}_{\xi}), u_{\sigma}, \mathbb{W}) \subseteq [f(x^*, u_{\sigma}, w^*) - D_x \delta_x - D_w \delta_w, f(x^*, u_{\sigma}, w^*) + D_x \delta_x + D_w \delta_w],$$

où, comme illustré dans la figure 3b, on ne calcule que le successeur de (1) correspondant aux centres des intervalles d'état et de perturbation, autour duquel on crée un intervalle de demi-largeur $D_x \delta_x + D_w \delta_w$ correspondant à la variation maximale du système après un pas de temps du fait des bornes sur ses dérivées.

Plusieurs autres méthodes d'analyse d'atteignabilité à base d'intervalles sont présentées dans [5] et implémentées dans l'outil Matlab TIRA [6]. Bien qu'il existe de nombreuses autres méthodes d'analyse d'atteignabilité utilisant des représentations géométriques plus complexes et générant des surapproximations plus précises, elles sont rarement utilisées au sein d'approches d'abstraction symbolique du fait de leur trop grande complexité par rapport aux méthodes à base d'intervalles. Les autres avantages de l'utilisation d'intervalles incluent :

- leur facilité d'implémentation, de manipulation et d'y vérifier l'inclusion d'un point ;
- la possibilité de partitionner l'espace d'état avec une grille uniforme ;
- l'intersection de deux intervalles est aussi un intervalle.

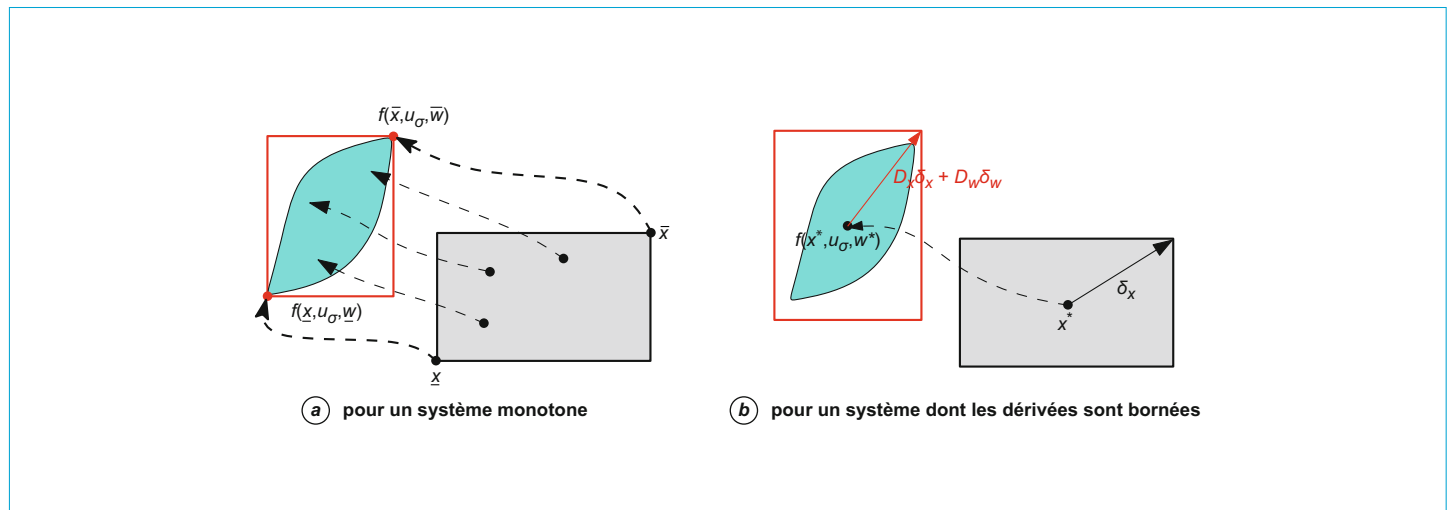


Figure 3 – Deux méthodes de surapproximation (intervalle rouge) de l'ensemble atteignable (bleu) à partir d'un intervalle d'états initiaux (gris)

Remarque 3

L'approche décrite dans cette section peut être appliquée pour tous les éléments de la partition \mathbb{X}_ξ avec $\xi \neq 0$. Pour $\xi = 0$, nous rappelons que $\mathbb{X}_0 = \mathbb{R}^{n_x} \setminus \mathbb{X}$. Il est d'usage dans ce cas particulier d'utiliser la surapproximation $f(\text{cl}(\mathbb{X}_0), u_\sigma, \mathbb{W}) \subseteq \mathbb{R}^{n_x}$.

1.2 Contrôleur symbolique et concrétisation

Considérons maintenant un contrôleur pour le modèle symbolique (2) de la forme :

$$\begin{cases} \psi(t) = h_1(\psi(t-1), \xi(t)), \\ \sigma(t) = h_2(\psi(t), \xi(t)) \end{cases} \quad (6)$$

où $\psi(t) \in \Psi$ est une variable symbolique interne du contrôleur et Ψ est un ensemble fini. Par convention $\psi(-1) = \psi_{\text{init}}$ où ψ_{init} est une valeur initiale. La forme du **contrôleur symbolique** (6) est suffisamment générale pour mettre en œuvre des spécifications variées et complexes, comme nous le verrons dans la section 2. Ce contrôleur possède une mémoire représentée par la variable interne $\psi(t)$ et est donc un contrôleur « dynamique ». Le cas d'un contrôleur « statique », c'est-à-dire sans mémoire, peut être vu comme un cas particulier où $h_2(\psi, \xi) = h(\xi)$. Dans ce cas, la commande symbolique $\sigma(t) = h(\xi(t))$ ne dépend que de l'état symbolique à l'instant t . On note $\mathcal{T}_{\text{symb}}(\xi_0)$ l'ensemble des trajectoires du système symbolique en boucle fermée (2)-(6) avec la condition initiale $\xi(0) = \xi_0$. Notons qu'il peut exister plusieurs trajectoires pour une condition initiale à cause du caractère multivalué de la fonction de transition g .

Nous décrivons maintenant la procédure de **concrétisation du contrôleur** (6) qui nous permettra de contrôler le système dynamique (1). Ainsi, nous considérons le contrôleur suivant :

$$\begin{cases} \psi(t) = h_1(\psi(t-1), q(x(t))), \\ u(t) = p(h_2(\psi(t), q(x(t)))) \end{cases} \quad (7)$$

où $q: \mathbb{R}^{n_x} \rightarrow \Xi$ et $p: \Sigma \rightarrow \mathbb{R}^{n_u}$ sont les interfaces d'abstraction et de concrétisation définies en section 1.1.1. On note $\mathcal{T}_{\text{conc}}(x_0)$ l'ensemble des trajectoires du système en boucle fermée (1)-(7) avec la condition initiale $x(0) = x_0$. En raison de la présence de la perturbation $w(\cdot)$, il peut exister plusieurs trajectoires pour une condition initiale. On note aussi :

$$q(\mathcal{T}_{\text{conc}}(x_0)) = \{\xi(\cdot) \mid \xi(\cdot) = q(x(\cdot)), x(\cdot) \in \mathcal{T}_{\text{conc}}(x_0)\}.$$

Cet ensemble correspond aux trajectoires du système en boucle fermée (1)-(7), observées via l'interface d'abstraction.

Nous pouvons maintenant énoncer le théorème fondamental du contrôle symbolique [2] (voir également figure 4).

Théorème 1

Pour tout $x_0 \in \mathbb{R}^{n_x}$, on a $q(\mathcal{T}_{\text{conc}}(x_0)) \subseteq \mathcal{T}_{\text{symb}}(q(x_0))$.

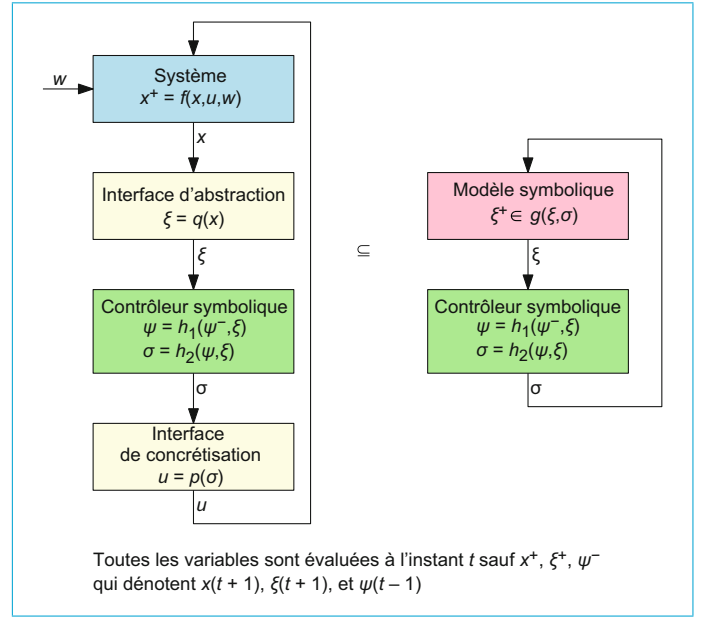


Figure 4 – Architecture de contrôle symbolique et illustration du théorème fondamental du contrôle symbolique

Intuitivement, le théorème 1 stipule que les trajectoires du système en boucle fermée (1)-(7) forment un sous-ensemble des trajectoires du système symbolique en boucle fermée (2)-(6). Cela implique que si un contrôleur symbolique (6) synthétisé comme dans la section 2 permet au modèle symbolique en boucle fermée de satisfaire des spécifications données, alors le contrôleur concrétisé (7) permettra aussi au système initial en boucle fermée de satisfaire les mêmes spécifications. Notez toutefois que l'échec de synthèse d'un contrôleur symbolique n'implique pas pour autant que le système initial est incapable de satisfaire les spécifications (contrairement aux abstractions complètes évoquées dans la section 4.2). En effet, puisque l'étape d'abstraction symbolique s'appuie sur des méthodes de surapproximation du système (1), cet échec peut être dû au fait d'avoir créé un modèle symbolique trop grossier pour capturer les comportements importants du système, ou à la présence de trop nombreuses transitions d'un état symbolique vers lui-même qui empêchent la progression du modèle symbolique vers les objectifs de contrôle. Ces problèmes peuvent parfois être résolus en adaptant les paramètres de discrétisation du système dans la section 1.1.1 (voir remarque 1 et section 4.1.1).

À retenir

- Un modèle symbolique est un système dynamique dont les ensembles d'états et de commandes sont des ensembles finis, et qui représente une « abstraction » d'un système dynamique continu.
- Un modèle symbolique peut être calculé par partition de l'espace des états, discrétisation de l'espace des commandes, et surapproximation des ensembles atteignables. Cette approche est applicable à des systèmes dynamiques complexes, non linéaires, avec des contraintes sur les états et les commandes, et des perturbations bornées.
- Un contrôleur symbolique peut être « concrétisé » de manière à ce que l'ensemble des trajectoires du système continu en boucle fermée soit inclus dans l'ensemble des trajectoires du modèle symbolique en boucle fermée.

2. Synthèse d'un contrôleur symbolique

Dans cette section, nous présentons des méthodes de synthèse d'un contrôleur permettant de garantir que le modèle symbolique satisfasse des spécifications données (étape 2 de *Synthèse* dans la figure 1). Nous considérons qu'une spécification est définie formellement par un sous-ensemble de trajectoires $S \subseteq \mathcal{F}(\mathbb{N}, \Xi)$, où $\mathcal{F}(\mathbb{N}, \Xi)$ dénote l'ensemble des fonctions de \mathbb{N} vers Ξ . Ainsi, le problème de synthèse de contrôleur peut être formulé de la manière suivante : étant donné un modèle symbolique (2) et une spécification S , synthétiser un contrôleur (6) et un ensemble d'états initiaux $Q_0 \subseteq \Xi$ tels que l'ensemble des trajectoires du modèle symbolique en boucle fermée satisfasse $\mathcal{T}_{\text{symp}}(\xi_0) \subseteq S$, pour tout $\xi_0 \in Q_0$.

Nous commencerons par une présentation détaillée de la synthèse d'un contrôleur symbolique dans le cas de spécifications de sûreté, correspondant aux objectifs de contrôle les plus communs dans le cadre des méthodes à base d'abstraction. Nous évoquerons ensuite brièvement comment adapter cette méthode de synthèse pour des spécifications d'atteignabilité et des spécifications plus complexes représentées par des automates.

Définissons ici l'opérateur de prédécesseurs $\text{Pre} : 2^\Xi \rightarrow 2^\Xi$ (où 2^Ξ dénote l'ensemble de tous les sous-ensembles de Ξ) qui joue un rôle clé dans la majorité des algorithmes de synthèse de contrôleur symbolique présentés dans cette section. Pour un sous-ensemble d'états symboliques $R \subseteq \Xi$:

$$\text{Pre}(R) = \{\xi \in \Xi \mid \exists \sigma \in \Sigma, 0 \neq g(\xi, \sigma) \subseteq R\}. \quad (8)$$

$\text{Pre}(R)$ représente ainsi l'ensemble des états symboliques pouvant atteindre l'ensemble R après une transition du modèle symbolique, c'est-à-dire que l'état $\xi \in \text{Pre}(R)$ est un prédécesseur de l'ensemble R s'il existe une commande symbolique σ pouvant être utilisée depuis l'état ξ ($g(\xi, \sigma) \neq \emptyset$) et dont tous les successeurs sont inclus dans l'ensemble R ($g(\xi, \sigma) \subseteq R$). Notons que par construction, les modèles symboliques construits en section 1.1.2 vérifient $g(\xi, \sigma) \neq \emptyset$, pour tout ξ et σ .

2.1 Propriétés de sûreté

Dans le cadre d'un problème de sûreté, il nous est donné un sous-ensemble $Q_s \subseteq \Xi$ d'états symboliques dans lequel on souhaite maintenir l'état du modèle symbolique. On définit alors la spécification de sûreté correspondante comme l'ensemble des trajectoires symboliques restant dans l'ensemble Q_s à tout instant :

$$S = \{\xi : \mathbb{N} \rightarrow \Xi \mid \forall t \in \mathbb{N}, \xi(t) \in Q_s\}.$$

2.1.1 Domaine de sûreté

La première étape de la synthèse d'un contrôleur de sûreté consiste en la détermination du domaine des contrôleurs de sûreté, c'est-à-dire le sous-ensemble $R^* \subseteq Q_s$ d'états symboliques sur lequel un contrôleur de sûreté peut être défini. Cette première étape est décrite par l'algorithme 1 dans lequel on construit itérativement les ensembles R_k jusqu'à ce qu'ils atteignent un point fixe correspondant au domaine de sûreté recherché.

L'ensemble R_0 est tout d'abord initialisé à Q_s . Ensuite, le prochain ensemble R_{k+1} est défini comme l'intersection entre l'ensemble visé Q_s (puisque tous les états $\xi \in \Xi \setminus Q_s$ sont automatiquement considérés comme non sûrs) avec l'ensemble des prédécesseurs de R_k défini dans (8). Autrement dit, un état symbolique de R_k est conservé dans R_{k+1} s'il existe au moins une commande symbolique pour laquelle tous les successeurs restent dans l'ensemble R_k ; et inversement, l'état n'est pas

transféré dans R_{k+1} si chacune de ses commandes symboliques a une chance de générer un successeur en dehors de R_k . Intuitivement, R_k représente donc l'ensemble des états initiaux à partir desquels il est possible de conserver la trajectoire dans l'ensemble Q_s pour au moins k pas de temps. Ces itérations sont répétées jusqu'à l'obtention d'un point fixe $R^* \subseteq Q_s$, qui correspond au domaine maximal d'un contrôleur de sûreté. De plus, du fait du caractère fini du modèle symbolique, nous avons la garantie que ce point fixe sera atteint en un nombre fini d'itérations. Un exemple pédagogique de l'application de l'algorithme 1 est présenté ci-après.

Algorithme 1 – Synthèse du domaine $R^* \subseteq Q_s$ des contrôleurs de sûreté

Entrées : $Q_s \subseteq \Xi$

$R_0 = Q_s$

répéter

$R_{k+1} = Q_s \cap \text{Pre}(R_k)$

jusqu'à $R_{k+1} = R_k$;

Sorties : $R^* = R_k$

Dans cet exemple, le modèle symbolique a six états ($\Xi = \{1, \dots, 6\}$), deux commandes ($\Sigma = \{a, b\}$), et certains états ne peuvent utiliser qu'une seule des deux commandes : $g(2, b) = g(4, b) = g(5, b) = \emptyset$. Notez toutefois que ce dernier choix est fait uniquement dans l'intérêt de cet exemple illustratif, puisque par construction, les modèles symboliques créés comme dans la section 1.1 vérifient $g(\xi, \sigma) = \emptyset$, pour tout ξ et σ . L'objectif de sûreté est d'éviter l'état symbolique 5, ce qui implique la définition de $Q_s = \{1, 2, 3, 4, 6\}$. Nous décrivons ci-dessous chaque itération de l'algorithme 1, initialisé avec $R_0 = Q_s$.

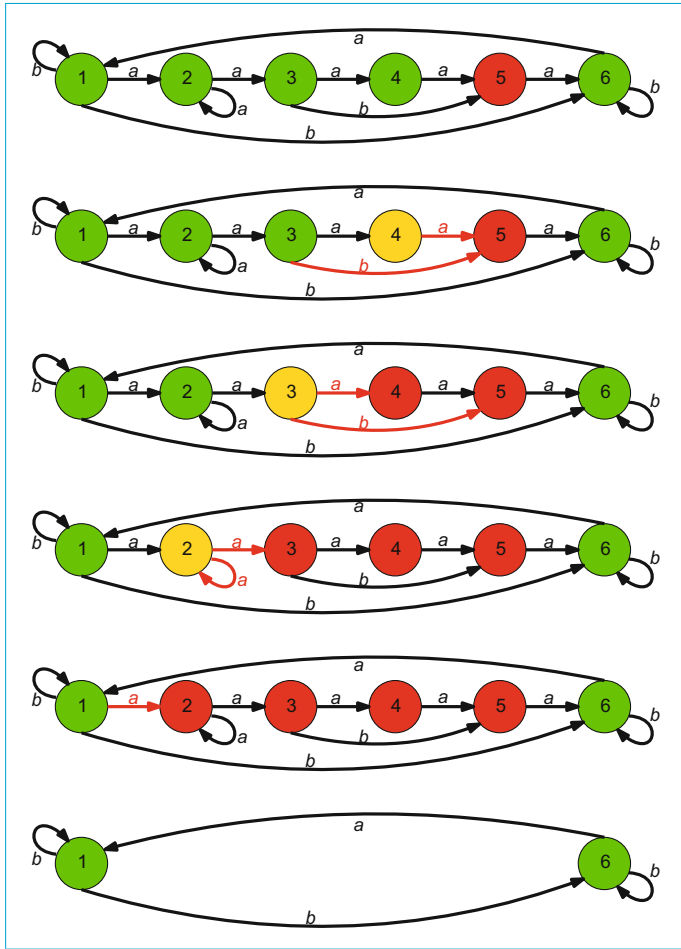
Dans la première itération, toutes les transitions partant des états 1, 2 et 6 vont vers R_0 et ces états sont donc conservés dans R_1 . Pour l'état 3, la commande b est considérée comme non sûre à cause de la transition $g(3, b) = 5 \notin R_0$ qui sort de R_0 , mais il existe une autre commande (a) qui est sûre ($g(3, a) = 4 \in R_0$) et qui permet donc à l'état 3 d'être aussi conservé dans R_1 . Enfin, on voit que l'état 4 n'est pas sûr car il ne peut pas utiliser la commande b ($g(4, b) = \emptyset$) et l'utilisation de la commande a ne permet pas de rester dans R_0 ($g(4, a) = 5 \notin R_0$). Nous obtenons alors $R_1 = R_0 \setminus \{4\} = \{1, 2, 3, 6\}$.

À la deuxième itération, on observe que depuis l'état 3, toutes les commandes mènent le système en dehors de R_1 ($g(3, a) = 4, g(3, b) = 5$). Il faut donc retirer cet état : $R_2 = R_1 \setminus \{3\} = \{1, 2, 6\}$.

À la troisième itération, on voit que l'état 2 n'est pas sûr car il ne peut pas utiliser la commande b ($g(2, b) = \emptyset$) et l'utilisation de la commande a a plusieurs successeurs possibles dont l'intersection avec les états à éviter est non vide ($g(2, a) = \{2, 3\} \not\subseteq R_2$). Cet état est donc retiré : $R_3 = R_2 \setminus \{2\} = \{1, 6\}$.

À la quatrième itération, on observe que la commande a est non sûre depuis l'état 1 ($g(1, a) = 2 \notin R_3$), mais la commande b est sûre depuis ce même état ($g(1, b) = \{1, 6\} \subseteq R_3$), ce qui nous permet de garder l'état 1 dans R_4 . Toutes les transitions partant de l'état 6 permettent aussi de rester dans R_3 . Cela signifie que l'algorithme a atteint un point fixe ($R_4 = R_3$) et se termine donc avec le domaine de sûreté $R^* = \{1, 6\}$.

Les contrôleurs de sûreté h possibles sont représentés dans la dernière illustration, où l'état 1 n'a qu'une commande autorisée ($h(1) = b$), et l'état 6 peut utiliser les deux commandes ($h(6) \in \{a, b\}$).



2.1.2 Contrôleurs sûrs

La structure du contrôleur symbolique défini dans (6) permet de couvrir une large gamme de spécifications complexes comme celles qui sont évoquées dans la section 2.2. Cependant, dans le cas d'une spécification de sûreté, la présence d'une mémoire (dépendance en $\psi(t-1)$) et celle de la variable interne ψ ne sont pas nécessaires. Un contrôleur symbolique de sûreté peut ainsi être défini par la structure simplifiée : $\sigma(t) = h(\xi(t))$ ou $\sigma = h(\xi)$, correspondant donc à un contrôleur statique par retour d'état.

Théorème 2

Si $R^* \neq \emptyset$, le problème de sûreté est résolu sur le modèle symbolique (2) par le contrôleur multivalué $H : \Xi \rightarrow 2^\Sigma$ défini par :

$$\forall \xi \in R^*, H(\xi) = \{\sigma \in \Sigma \mid 0 \neq g(\xi, \sigma) \subseteq R^*\}.$$

Ainsi, l'utilisation en boucle fermée sur le modèle symbolique de tout contrôleur $h : \Xi \rightarrow \Sigma$ satisfaisant $h(\xi) \in H(\xi)$ pour tout $\xi \in R^*$ garantit alors la satisfaction de la spécification de sûreté :

$$\forall \xi_0 \in R^*, \mathcal{T}_{\text{symb}}(\xi_0) \subseteq \{\xi : \mathbb{N} \rightarrow \Xi \mid \forall t \in \mathbb{N}, \xi(t) \in Q_s\}.$$

Une fois qu'un contrôleur sûr $h : \Xi \rightarrow \Sigma$ a été synthétisé pour le modèle symbolique, nous pouvons le concrétiser comme dans (7)

en un contrôleur sûr pour le système concret (1). La sûreté du contrôleur concrétisé est garantie par le théorème 1.

Corollaire 1

Si $R^* \neq \emptyset$ et $h : \Xi \rightarrow \Sigma$ est défini comme dans le théorème 2, alors le contrôleur concrétisé

$$u(t) = p(h(q(x(t))))$$

garantit que le système concret (1) en boucle fermée satisfait la spécification de sûreté :

$$\forall x_0 \in \mathbb{R}^{n_x} \text{ tel que } q(x_0) \in R^*,$$

$$q(\mathcal{T}_{\text{conc}}(x_0)) \subseteq \{\xi : \mathbb{N} \rightarrow \Xi \mid \forall t \in \mathbb{N}, \xi(t) \in Q_s\}.$$

En conclusion, si l'algorithme 1 atteint un point fixe R^* non vide, il nous est alors possible de définir des contrôleurs (dont la définition est proche des conditions de l'opérateur Pre) pour garantir la satisfaction en boucle fermée des spécifications de sûreté sur le modèle symbolique (théorème 2) et le système concret (1) (corollaire 1). En revanche, obtenir un domaine de sûreté vide ($R^* = \emptyset$) dans l'algorithme 1 n'est pas une garantie que le problème de sûreté ne peut pas être résolu sur le système concret (1). En effet, $R^* = \emptyset$ indique uniquement que la spécification de sûreté ne peut pas être satisfaite sur le modèle symbolique considéré. Mais ce modèle ayant été créé à partir de surapproximation des dynamiques concrètes de (1), il reste possible que le problème de sûreté puisse être résolu via une procédure d'abstraction symbolique moins grossière (en prenant une discrétisation plus fine des états et commandes, ou en utilisant des méthodes moins conservatives pour la surapproximation des dynamiques concrètes).

2.1.3 Optimisation des contrôleurs

Suite à l'application de l'algorithme 1, le caractère multivalué du contrôleur de sûreté $H : \Xi \rightarrow 2^\Sigma$ synthétisé dans le théorème 2 nous permet d'introduire des critères de performance pour optimiser le choix des commandes sûres que l'on associe à chaque état symbolique.

Dans cette section, nous considérons une trajectoire du système concret (1) en boucle fermée $(x_0, x_1, \dots) \in \mathcal{T}_{\text{conc}}(x_0)$ associée à une séquence de commandes (u_0, u_1, \dots) telle que $u_t \in p(H(q(x_t)))$ pour tout $t \in \mathbb{N}$, pour lesquelles nous souhaitons minimiser le critère de performance

$$\sum_{t=0}^{+\infty} \lambda^t j(x_t, u_t), \quad (9)$$

où $j(x, u)$ est le coût lié à l'utilisation de la commande u à partir de l'état x , et $\lambda \in (0, 1)$ est un facteur de décompte permettant de réduire l'influence des incertitudes dues aux perturbations dans les étapes futures. Afin de réaliser l'optimisation dans le domaine symbolique à l'aide d'un algorithme de programmation dynamique, on considère une approximation de ce critère de performance tronqué sur un horizon temporel fini de N pas de temps :

$$\sum_{t=0}^N \lambda^t j_{\text{symb}}(\xi_t, \sigma_t), \quad (10)$$

où l'on définit $j_{\text{symb}}(\xi, \sigma) = \max_{x \in \mathbb{X}_\xi} j(x, u_\sigma)$.

L'algorithme de programmation dynamique [7] utilisé pour minimiser (10) est une procédure calculant de manière itérative les valeurs de l'opérateur J^t , correspondant à l'accumulation des coûts j_{symb} le long d'une trajectoire symbolique en boucle fermée entre les instants t et la fin de l'horizon temporel N . Puisque le

modèle symbolique, même en boucle fermée avec un contrôleur de sûreté, a une fonction de transition multivaluée, cette minimisation est réalisée sur le pire coût des trajectoires possibles (puisque l'incertitude des successeurs atteints est subie et non contrôlée).

$$J^N(\xi) = \min_{\sigma^N \in H(\xi)} j_{\text{symb}}(\xi, \sigma^N) \quad (11)$$

$$J^t(\xi) = \min_{\sigma^t \in H(\xi)} \left(j_{\text{symb}}(\xi, \sigma^t) + \lambda \max_{\xi' \in g(\xi, \sigma^t)} J^{t+1}(\xi') \right) \quad (12)$$

$$h^*(\xi) \in \arg \min_{\sigma^0 \in H(\xi)} \left(j_{\text{symb}}(\xi, \sigma^0) + \lambda \max_{\xi' \in g(\xi, \sigma^0)} J^1(\xi') \right) \quad (13)$$

Après le calcul de toutes les itérations de (11)-(12), on obtient $J^0(\xi)$ minimisant (10) face au pire cas des trajectoires partant de ξ , et cette minimisation est associée à une séquence de commandes symboliques $(\sigma^0, \dots, \sigma^N)$ à utiliser à chaque pas de temps de l'horizon temporel. Puisque cette stratégie de commande n'est optimale depuis l'état ξ que dans le cas où l'incertitude des transitions multivaluées force le système à suivre les transitions aux coûts les plus élevés, il est préférable (du point de vue des performances du système en boucle fermée) d'implémenter une stratégie de contrôle à fenêtre glissante (13) dans laquelle à chaque instant on mesure l'état symbolique actuel ξ , applique la première commande $h^*(\xi) = \sigma^0$ de la séquence obtenue ci-dessus, puis recommence ces deux étapes lors de la mesure de l'état symbolique à l'instant suivant [8]. Notez que tous les calculs (11)-(13) pour obtenir J^0 et h^* ne sont réalisés qu'une seule fois à l'avance, et seules les deux étapes de mesure de ξ et d'application de la commande optimal $h^*(\xi)$ sont réalisées en temps réel.

Bien que l'optimisation des performances (11)-(13) ait été réalisée dans le domaine symbolique et sur un horizon de temps fini (10), il est tout de même possible d'en déduire des garanties sur le critère de performance (9) pour les trajectoires infinies du système concret (1).

Théorème 3

Soit $(x^0, x^1, \dots) \in \mathcal{T}_{\text{conc}}(x^0)$ avec $q(x^0) \in R^*$ une trajectoire de (1) utilisant le contrôleur $p \circ h^* \circ q : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}$ ($u_t = p(h^*(q(x_t)))$ pour tout $t \in \mathbb{N}$). Alors, on a :

$$\sum_{t=0}^{+\infty} \lambda^t j(x^t, u^t) \leq J^0(q(x^0)) + \frac{\lambda^{N+1}}{1-\lambda} \max_{\xi \in R^*} \min_{\sigma \in H(\xi)} j_{\text{symb}}(\xi, \sigma).$$

Ce résultat signifie que pour la trajectoire de $\mathcal{T}_{\text{conc}}(x^0)$ sur laquelle le critère de performance (9) est calculé, il est possible de fournir une borne supérieure à la valeur de ce critère. Sa première partie $J^0(q(x^0))$ est la minimisation dans le pire cas des performances symboliques (10) sur un horizon fini de N pas de temps. La seconde partie est une constante indépendante de la trajectoire suivie qui représente le coût du reste de la trajectoire à partir de l'instant $N+1$.

L'approche présentée dans cette section, issue de [9], n'est qu'une possibilité permettant de profiter du caractère multivalué du contrôleur de sûreté H pour optimiser un critère de performance. Une alternative présentée dans [10] utilise une optimisation en temps réel sous la forme d'un algorithme de commande prédictive. Dans cette approche, des contraintes terminales sont ajoutées à la formulation du problème d'optimisation sous la forme d'un ensemble invariant contrôlé permettant d'assurer la faisabilité récursive du problème. Les ensembles invariants contrôlés étant généralement difficiles à calculer pour les systèmes non linéaires (1), l'approche proposée utilise la création d'un modèle symbolique et la synthèse d'un contrôleur de sûreté pour calculer

le domaine de sûreté R^* correspondant à l'ensemble invariant contrôlé désiré.

2.2 Autres spécifications

2.2.1 Propriétés d'atteignabilité

Une propriété d'atteignabilité consiste à conduire les trajectoires du modèle symbolique en boucle fermée vers un sous-ensemble cible $Q_a \subseteq \Xi$. Formellement, une spécification d'atteignabilité est donnée par l'ensemble de trajectoires suivant :

$$S = \{ \xi : \mathbb{N} \rightarrow \Xi \mid \exists t \in \mathbb{N}, \xi(t) \in Q_a \}.$$

La synthèse de contrôleurs réalisant la spécification d'atteignabilité est similaire à celle des contrôleurs de sûreté. En effet, considérons la séquence d'ensembles R_k calculée par l'algorithme 2. Notons que l'unique différence avec l'algorithme 1 est le remplacement du calcul de l'intersection par celui de l'union. L'ensemble R_k représente l'ensemble des états initiaux à partir desquels on peut amener les trajectoires du modèle symbolique à atteindre Q_a en au plus k pas de temps.

Algorithme 2 – Synthèse du domaine $R^* \supseteq Q_a$ des contrôleurs d'atteignabilité

Entrées : $Q_a \subseteq \Xi$

$R_0 = Q_a$

répéter

$R_{k+1} = Q_a \cup \text{Pre}(R_k)$

jusqu'à $R_{k+1} = R_k$;

Sorties : $R^* = R_k$

Une fois la séquence d'ensemble R_k obtenue par l'algorithme 2, un contrôleur d'atteignabilité statique peut être obtenu de la manière suivante :

Théorème 4

Le problème d'atteignabilité est résolu sur le modèle symbolique (2) par tout contrôleur $h : \Xi \rightarrow \Sigma$ satisfaisant :

$$\forall k \in \mathbb{N}, \forall \xi \in R_{k+1}, 0 \neq g(\xi, h(\xi)) \subseteq R_k.$$

L'utilisation de l'un de ces contrôleurs en boucle fermée sur le modèle symbolique garantit alors la satisfaction de la spécification :

$$\forall \xi_0 \in R^*, \mathcal{T}_{\text{symb}}(\xi_0) \subseteq \{ \xi : \mathbb{N} \rightarrow \Xi \mid \exists t \in \mathbb{N}, \xi(t) \in Q_a \}.$$

En conséquence des théorèmes 1 et 4, un résultat similaire au corollaire 1 peut être obtenu pour les propriétés d'atteignabilité.

2.2.2 Spécifications basées sur des automates

Les propriétés de sûreté et d'atteignabilité sont des spécifications élémentaires pour lesquelles des contrôleurs statiques par retour d'état peuvent être synthétisés. Toutefois, pour certaines spécifications plus complexes, des contrôleurs dynamiques peuvent s'avérer nécessaires. C'est par exemple le cas des spécifications basées sur des automates, décrites dans cette section.

Considérons une **fonction de labélisation** $l : \Xi \rightarrow L$ où L est un ensemble fini de **labels**. Nous définissons ensuite un automate d'états finis [11] \mathcal{A} caractérisé par un ensemble d'états Ψ , un état initial $\psi_{\text{init}} \in \Psi$, un ensemble d'états finaux $\Psi_f \subseteq \Psi$, l'ensemble de labels L et une fonction de transition $\delta : \Psi \times L \rightarrow \Psi$. Étant donnée une fonction $\xi \in \mathcal{F}(\mathbb{N}, \Xi)$, nous pouvons associer une **trace** de l'automate, notée $\psi = \tau_{\mathcal{A}}(\xi) \in \mathcal{F}(\mathbb{N}, \Psi)$ et définie par :

$$\psi(t) = \delta(\psi(t-1), l(\xi(t)))$$

avec $\psi(-1) = \psi_{\text{init}}$. La spécification définie par l'automate \mathcal{A} correspond alors à l'ensemble de trajectoires dont la trace atteint un état final :

$$S_{\mathcal{A}} = \{\xi : \mathbb{N} \rightarrow \Xi \mid \exists t \in \mathbb{N}, \psi(t) \in \Psi_f \text{ avec } \psi = \tau_{\mathcal{A}}(\xi)\}.$$

Un exemple illustratif de spécification basée sur un automate est présenté dans l'exemple ci-dessous.

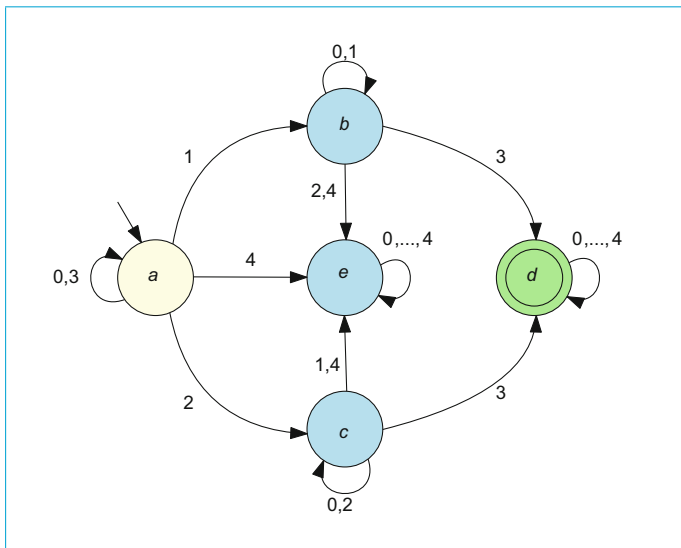
Dans cet exemple, on considère quatre sous-ensembles disjoints Q_1, Q_2, Q_3 et Q_4 de l'ensemble des états symboliques Ξ . Nous souhaitons formaliser la spécification définie dans le langage courant de la manière suivante.

« Allez en Q_1 ou en Q_2 , mais uniquement en l'un des deux, puis en Q_3 , tout en évitant Q_4 tout le long du chemin. »

Cette spécification peut être formellement décrite à l'aide d'un automate d'états finis \mathcal{A} avec :

- l'ensemble des labels $L = \{0, 1, 2, 3, 4\}$ et la fonction de labélisation définie par $l(\xi) = i$ si $\xi \in Q_i, i = 1, 2, 3, 4$ ou $l(\xi) = 0$ si $\xi \in \Xi \setminus (Q_1 \cup \dots \cup Q_4)$;
- l'ensemble des états de l'automate $\Psi = \{a, b, c, d, e\}$, l'état initial $\psi_{\text{init}} = a$ et l'ensemble d'états finaux $\Psi_f = \{d\}$;
- la fonction de transition δ décrite par le graphe sur la figure ci-dessous.

Ainsi, pour atteindre l'état final d et satisfaire la spécification $S_{\mathcal{A}}$, il faut d'abord aller en Q_1 (transition de a vers b) ou en Q_2 (transition de a vers c) en évitant Q_4 (transition de a vers e). Si Q_1 est atteint, il faut aller en Q_3 (transition de b vers d) en évitant Q_2 et Q_4 (transition de b vers e). Inversement, si Q_2 est atteint, il faut aller en Q_3 (transition de c vers d) en évitant Q_1 et Q_4 (transition de c vers e). L'état e est un état puits à partir duquel il n'est plus possible d'atteindre l'état final d et donc de satisfaire la spécification $S_{\mathcal{A}}$.



La synthèse d'un contrôleur pour la spécification $S_{\mathcal{A}}$ peut être réalisée en suivant la méthodologie suivante : considérons un contrôleur dynamique de la forme (6) où l'ensemble des états internes du contrôleur coïncide avec l'ensemble des états Ψ de l'automate \mathcal{A} . On considère également la fonction $h_1 : \Psi \times \Xi \rightarrow \Psi$ définie pour tout ψ et ξ par $h_1(\psi, \xi) = \delta(\psi, l(\xi))$. On peut alors définir le modèle symbolique étendu :

$$\begin{cases} \psi(t+1) = h_1(\psi(t), \xi(t+1)), \\ \xi(t+1) \in g(\xi(t), \sigma(t)) \end{cases} \quad (14)$$

avec $\psi(0) = h_1(\psi_{\text{init}}, \xi(0))$. En notant $\tilde{\xi}(t) = \begin{pmatrix} \psi(t) \\ \xi(t) \end{pmatrix}$, le système (14) peut être réécrit sous une forme similaire à (2) :

$$\tilde{\xi}(t+1) \in \tilde{g}(\tilde{\xi}(t), \sigma(t)), \tilde{\xi}(t) \in \Psi \times \Xi, \sigma(t) \in \Sigma. \quad (15)$$

Par ailleurs, il peut être montré qu'une trajectoire ξ de (2) satisfait la spécification $S_{\mathcal{A}}$ si et seulement si la trajectoire correspondante de (15) satisfait la spécification d'atteignabilité :

$$\tilde{S}_{\mathcal{A}} = \{\tilde{\xi} : \mathbb{N} \rightarrow \Xi \mid \exists t \in \mathbb{N}, \tilde{\xi}(t) \in \Psi_f \times \Xi\}.$$

En suivant l'approche décrite dans la section 2.2.1, il est alors possible de synthétiser un contrôleur $\hat{h} : \Psi \times \Xi \rightarrow \Sigma$ et un ensemble d'états initiaux $\tilde{Q}_0 \subseteq \Psi \times \Xi$ pour la spécification $\tilde{S}_{\mathcal{A}}$. On peut alors définir la fonction $h_2 : \Psi \times \Xi \rightarrow \Sigma$ du contrôleur (6) par $h_2(\psi, \xi) = \hat{h}(\tilde{\xi})$, et l'ensemble d'états initiaux $Q_0 \subseteq \Xi$ par :

$$Q_0 = \{\xi \in \Xi \mid (h_1(\psi_{\text{init}}, \xi), \xi) \in \tilde{Q}_0\}$$

Théorème 5

Le problème de synthèse pour la spécification $S_{\mathcal{A}}$ est résolu sur le modèle symbolique (2) par le contrôleur (6) avec h_1 et h_2 définis ci-dessus. L'utilisation de ce contrôleur en boucle fermée sur le modèle symbolique garantit alors la satisfaction de la spécification définie par l'automate \mathcal{A} :

$$\forall \xi_0 \in Q_0, \mathcal{I}_{\text{symp}}(\xi_0) \subseteq S_{\mathcal{A}}.$$

En conséquence des théorèmes 1 et 5, un résultat similaire au corollaire 1 peut être obtenu pour les spécifications basées sur les automates.

2.2.3 Spécifications basées sur la logique temporelle

Il est possible d'utiliser les approches symboliques pour synthétiser des contrôleurs pour des spécifications encore plus complexes, telles que celles formalisées par des formules de logique temporelle. Les logiques temporelles sont construites à partir des opérateurs usuels de la logique classique auxquels s'ajoutent des opérateurs permettant de raisonner sur des séquences temporelles. Il est ainsi possible de caractériser à l'aide d'une formule de logique temporelle un sous-ensemble de trajectoires $\mathcal{S} \subseteq \mathcal{F}(\mathbb{N}, \Xi)$. La logique temporelle linéaire (LTL [12]) est la plus couramment utilisée pour formaliser des spécifications complexes pour des problèmes de contrôle. LTL est un formalisme riche de spécifications. Par exemple, les spécifications basées sur les automates d'états finis et décrites dans la section 2.2.2 correspondent à un petit fragment de LTL appelé « Co-Safe LTL ». Des automates plus complexes (tels que les automates de Büchi ou automates de Rabin) sont nécessaires pour résoudre le problème de synthèse de contrôleurs pour des spécifications LTL arbitraires. La méthodologie n'est pas exposée ici mais peut être trouvée dans [13].

À retenir

- Puisque le format des modèles symboliques est indépendant de la forme ou de la complexité du système dynamique continu, il nous est possible de proposer une structure unifiée pour la synthèse de contrôleurs symboliques qui ne dépend que des spécifications visées.
- Pour les spécifications élémentaires de sûreté ou d'atteignabilité, la synthèse de contrôleurs symboliques est réalisée via un nombre fini d'itérations sur les ensembles de transitions symboliques.
- Une large gamme de spécifications plus complexes (formules de logique temporelle) peut également être considérée en exprimant la spécification sous la forme d'un automate, puis en réalisant une synthèse d'atteignabilité sur le système résultant du produit entre le modèle symbolique et l'automate des spécifications.

3. Exemple numérique : contrôle d'un robot mobile

Dans cette section, nous présentons une illustration de l'approche symbolique pour la synthèse de contrôleur à l'aide d'un exemple inspiré d'un problème de robotique mobile. Les scripts Matlab permettant de reproduire l'exemple sont disponibles sur le dépôt :

<https://gitlab-research.centralesupelec.fr/antoine.girard/tutorial-example-on-symbolic-control>

Nota : les temps de calcul indiqués sont obtenus sur un ordinateur portable (logiciel : Matlab_R2019b ; processeur : 2,8 GHz Intel Core i7 quatre cœurs ; mémoire : 16 Go 2133 MHz LPDDR3).

3.1 Modèle et spécification

Considérons donc un robot mobile dans un plan, de type monocycle, dont l'état est caractérisé par sa position $(x_1, x_2) \in [0, x_1] \times [0, x_2]$ et par son cap $x_3 \in [-\pi, \pi]$. La dynamique en temps discret du système est décrite par les équations :

$$\begin{cases} x_1(t+1) = x_1(t) + \tau(u_1(t) \cos(x_3(t)) + w_1(t)) \\ x_2(t+1) = x_2(t) + \tau(u_1(t) \sin(x_3(t)) + w_2(t)) \\ x_3(t+1) = x_3(t) + \tau(u_2(t) + w_3(t)) \pmod{2\pi} \end{cases} \quad (16)$$

où $u_i(t) \in [\underline{u}_i, \bar{u}_i]$, $i = 1, 2$, sont les commandes du système représentant les vitesses linéaires et angulaires du robot, $w_i(t) \in [-\bar{w}_i, \bar{w}_i]$, $i = 1, 2, 3$ sont des perturbations, et τ est la période d'échantillonnage. Dans la suite, nous considérerons les valeurs numériques des paramètres présentées dans le tableau 1.

Nous décrivons maintenant la spécification du problème de contrôle considéré. On considère quatre régions disjointes R_1, R_2, R_3, R_4 du domaine $\mathbb{X} = [0, x_1] \times [0, x_2] \times [-\pi, \pi]$, illustrées dans la figure 5 et dont les valeurs numériques sont présentées dans le tableau 2. La spécification considérée peut être décrite dans le langage courant de la manière suivante :

« Allez en R_1 ou en R_2 , mais uniquement en l'un des deux, puis en R_3 , tout en évitant R_4 tout le long du chemin. »

Notons qu'il s'agit de la même spécification que celle considérée dans l'encadré de la section 2.2.2. Ainsi, nous avons affaire à une spécification de type Co-Safe LTL, pouvant être formalisée à l'aide d'un automate d'états finis.

Tableau 1 – Valeurs numériques des paramètres du système (16).

| | | |
|-------------------|-------|------|
| \bar{x}_1 | m | 10 |
| \bar{x}_2 | m | 10 |
| \underline{u}_1 | m/s | 0.25 |
| \bar{u}_1 | m/s | 1 |
| \underline{u}_2 | rad/s | -1 |
| \bar{u}_2 | rad/s | 1 |
| \bar{w}_1 | m/s | 0.05 |
| \bar{w}_2 | m/s | 0.05 |
| \bar{w}_3 | rad/s | 0.05 |
| τ | s | 1 |

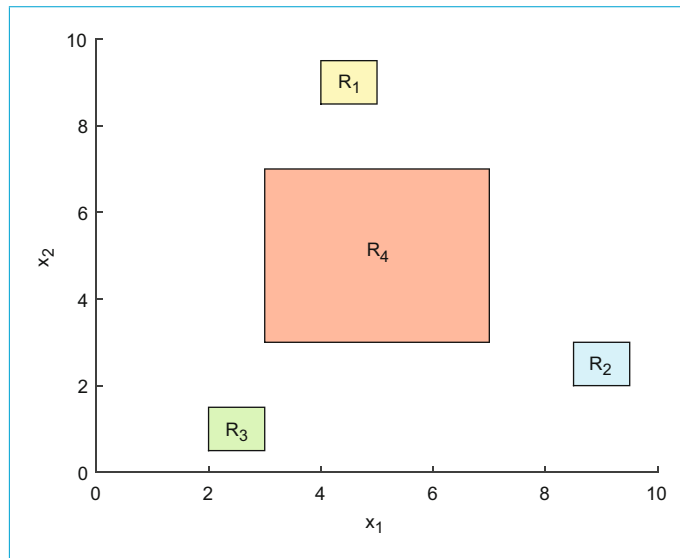


Figure 5 – Environnement du problème

Tableau 2 – Valeurs numériques des régions intervenant dans la spécification \mathcal{S} .

| | |
|-------|---|
| R_1 | $[4, 5] \times [8.5, 9.5] \times [-\pi, \pi]$ |
| R_2 | $[8.5, 9.5] \times [2, 3] \times [-\pi, \pi]$ |
| R_3 | $[2, 3] \times [0.5, 1.5] \times [-\pi, \pi]$ |
| R_4 | $[3, 7] \times [3, 7] \times [-\pi, \pi]$ |

3.2 Abstraction et synthèse de contrôleur

Nous suivons l'approche décrite dans la section 1 pour le calcul du modèle symbolique. Nous considérons une partition cartésienne de l'espace d'état \mathbb{X} obtenu en partitionnant uniformément

Tableau 3 – Valeurs numériques des paramètres de discrétisation.

| | |
|-----------|-----|
| N_{x_1} | 100 |
| N_{x_2} | 100 |
| N_{x_3} | 30 |
| N_{u_1} | 3 |
| N_{u_2} | 5 |

les intervalles $[0, \bar{x}_1]$, $[0, \bar{x}_2]$, $[-\pi, \pi]$ en N_{x_1} , N_{x_2} et N_{x_3} sous-intervalles. Pour la discrétisation des commandes, nous utilisons également des discrétisations uniformes des intervalles $[u_1, \bar{u}_1]$ et $[u_2, \bar{u}_2]$ en N_{u_1} et N_{u_2} valeurs de l'intervalle. Les valeurs retenues pour notre exemple sont montrées dans le tableau 3. Notons que ce choix résulte en un modèle symbolique avec 300001 états symboliques et 15 commandes symboliques.

Pour le calcul de la fonction de transition du modèle symbolique, nous utilisons une méthode de surapproximation de l'ensemble atteignable basée sur les dérivées bornées et décrite dans la section 2.2.1. Pour cela, nous utilisons les bornes suivantes sur les dérivées partielles de f :

$$\left| \frac{\partial f}{\partial x}(x, u, w) \right| \leq D_x = \begin{pmatrix} 1 & 0 & \tau |u_1| \\ 0 & 1 & \tau |u_1| \\ 0 & 0 & 1 \end{pmatrix}, \left| \frac{\partial f}{\partial w}(x, u, w) \right| \leq D_w = \begin{pmatrix} \tau & 0 & 0 \\ 0 & \tau & 0 \\ 0 & 0 & \tau \end{pmatrix}.$$

Le calcul du modèle symbolique par notre script Matlab nécessite un temps de calcul de l'ordre de 3 minutes.

Pour la synthèse de contrôleur, on considère la spécification symbolique présentée dans l'encadré de la section 2.2.2 où les ensembles symboliques Q_1 , Q_2 , Q_3 , Q_4 sont définis comme :

$$Q_i = \{ \xi \in \Xi \mid \mathbb{X}_\xi \subseteq R_i \}, \quad i = 1, \dots, 4.$$

La synthèse du contrôleur symbolique peut ensuite être réalisée par la méthode présentée dans la section 2.2.2. Notons que le contrôleur obtenu est un contrôleur dynamique par retour d'état de la forme (6). Le calcul du contrôleur symbolique par notre script Matlab nécessite un temps de calcul de l'ordre de 9 minutes. Le point fixe de l'algorithme de synthèse de contrôleur est atteint après 35 itérations.

3.3 Simulations numériques

Le contrôleur symbolique est concrétisé via la relation (7) et utilisé dans le cadre de simulations numériques pour validation. Sur la figure 6, nous représentons deux trajectoires obtenues pour deux conditions initiales distinctes. Nous pouvons voir que la spécification est satisfaite : la première trajectoire (en bleu) passe par R_1 puis atteint R_3 sans passer par R_2 ; la seconde trajectoire (en rouge) passe par R_2 puis atteint R_3 sans passer par R_1 . Dans les deux cas, R_4 est évité tout le long de la trajectoire. Sur la figure 7, nous représentons 50 trajectoires obtenues pour la même condition initiale et 50 différentes réalisations de la perturbation. Nous pouvons constater que la spécification est satisfaite dans tous les cas. Par ailleurs, notons que le comportement qualitatif des trajectoires n'est pas toujours le même. En effet, certaines trajectoires passent par R_1 alors que d'autres passent

par R_2 . Cela démontre que le contrôleur synthétisé est capable d'adapter le chemin suivi en fonction des incertitudes de la dynamique liées aux perturbations.

À retenir

- Sur un exemple inspiré d'un problème de robotique mobile, l'approche symbolique permet de synthétiser de manière automatique un contrôleur « correct par construction » pour une spécification non triviale.
- Les simulations numériques confirment la validité du contrôleur synthétisé et démontrent ses capacités d'adaptation face aux incertitudes sur la dynamique du système.

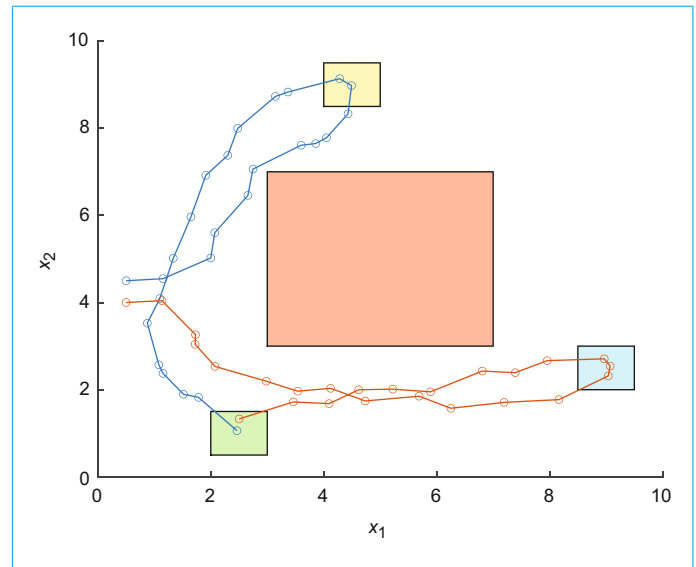


Figure 6 – Trajectoires du système en boucle fermée pour deux conditions initiale différentes : (0.5, 4.5, 0) ; (0.5, 4, 0)

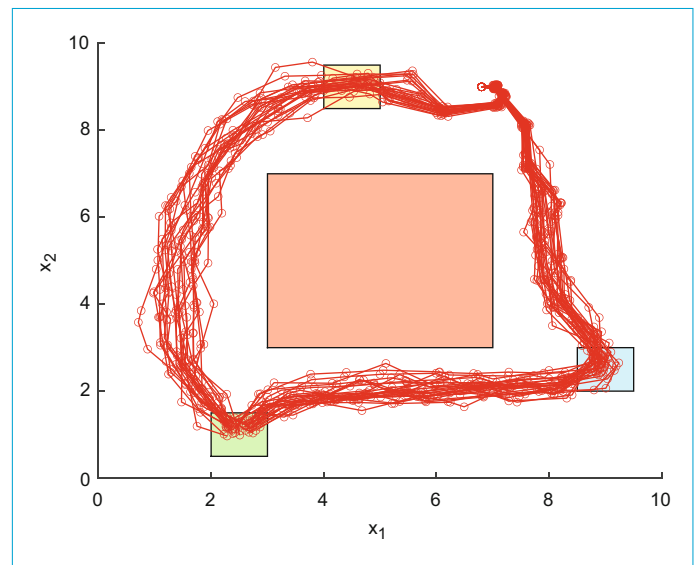


Figure 7 – Trajectoires du système en boucle fermée pour une condition initiale (6.8, 9, 0) et 50 réalisations de la perturbation

4. Thèmes avancés en commande symbolique

Les approches de contrôle symbolique sont aujourd'hui considérées comme un outil puissant pour le contrôle des systèmes dynamiques. Elles présentent plusieurs avantages par rapport aux approches classiques de l'automatique :

- traitement d'une large classe de systèmes (non linéaires, hybrides, incertains...) sous contraintes ;
- traitement de spécifications complexes comme celles évoquées dans la section 2. Ces spécifications sont très représentatives des systèmes dynamiques modernes tels que les robots ou les véhicules autonomes ;
- prise en compte de l'interaction entre le logiciel de contrôle et les processus physiques : le modèle symbolique et le logiciel de contrôle implémenté dans la plateforme de calcul numérique sont décrits dans un cadre unifié (par exemple, en tant que dynamiques symboliques). Cela permet de tenir compte des contraintes sur la partie logiciel lors de la synthèse du contrôleur ;
- l'approche est automatique et formelle : la synthèse du contrôleur est effectuée de manière automatique, sans recourir à des techniques heuristiques (contrairement à des contrôleurs PID, qui nécessitent des réglages et des tests). De plus, les garanties sont formelles, au sens où le système en boucle fermée satisfait la spécification. Ce type de garantie est obtenu en combinant deux types de preuves : des preuves mathématiques dans la phase d'abstraction, en se basant par exemple sur des outils d'analyse d'atteignabilité ; et des preuves algorithmiques, durant la phase de synthèse de contrôleur en se basant sur des outils de la communauté des méthodes formelles tels que les algorithmes de vérification de modèles [12] ou les démonstrations automatiques de théorèmes [14] ;
- tous les calculs peuvent être effectués hors ligne, et donc l'implémentation du contrôleur obtenu à l'étape de synthèse correspond uniquement à une table de correspondance ;
- l'approche symbolique est robuste, dans le sens où elle permet de prendre en considération différents types de perturbations. Nous avons expliqué dans la section 1 comment l'approche symbolique peut prendre en considération des perturbations bornées sur le modèle. L'approche symbolique offre la possibilité de considérer divers types de perturbations [15] [16], tels que les erreurs de mesure, généralement attribuables à l'utilisation de capteurs bruités ou à des algorithmes d'estimation (observateurs, filtres de Kalman).

Néanmoins, en dépit de ces avantages indéniables, le domaine du contrôle symbolique n'est pas dépourvu de défis et de limitations. Ces défis incluent des aspects tels que le passage à l'échelle, la complétude et la dépendance envers un modèle mathématique. Dans les sections suivantes, nous examinons ces problématiques, détaillant les enjeux spécifiques, tout en explorant les approches phares décrites dans la littérature pour les surmonter.

4.1 Abstractions efficaces et compositionnelles

Les modèles symboliques sont généralement obtenus par discrétisation des espaces d'état et de commande. En raison de la discrétisation, ces techniques d'abstraction souffrent du fléau de la dimensionnalité (le nombre d'états symboliques, respectivement de commandes, augmente exponentiellement par rapport à la dimension de l'espace d'état, respectivement de commande), ce qui rend ces approches difficiles à appliquer pour les systèmes à grande échelle. Dans ce qui suit, nous allons présenter les diffé-

rentes approches qui ont été proposées dans la littérature pour renforcer le passage à l'échelle des techniques de contrôle symbolique.

4.1.1 Abstractions multiéchelles

Dans ce type d'approche, une abstraction grossière du système est initialement construite. Le modèle symbolique ainsi obtenu est ensuite raffiné localement de manière itérative (par le biais d'une redécomposition de l'espace d'état) dans les régions empêchant la satisfaction de la spécification. Cette approche a été proposée dans la littérature pour des systèmes à commutation dans [17] ainsi que pour des systèmes non linéaires dans [18]. Une illustration de ce type d'approche est donnée dans la figure 8.

4.1.2 Abstractions paresseuses

Dans l'approche classique du contrôle symbolique, illustrée sur la figure 1, la première étape d'abstraction symbolique est suivie par la synthèse du contrôleur. Les abstractions paresseuses représentent une approche alternative dans laquelle on ne calcule que le fragment du modèle symbolique nécessaire à la synthèse du contrôleur. Dans ce type d'approche, les phases d'abstraction et de synthèse de contrôleur ne sont pas disjointes mais sont toutes les deux exécutées en parallèle. Ce type d'approche peut être utilisé pour des systèmes linéaires comme c'est le cas dans [19]. L'exploitation de certaines propriétés structurales du système permet en général d'améliorer les performances des approches d'abstraction paresseuse, comme c'est le cas dans [20] (respectivement, [21]), où les auteurs exploitent la propriété de monotonie (respectivement, symétrie) afin d'améliorer le temps de calcul pour l'abstraction et la synthèse de contrôleurs. Les approches d'abstraction paresseuse peuvent également être combinées avec les abstractions multiéchelles comme c'est le cas dans [22] pour des spécifications de sûreté.

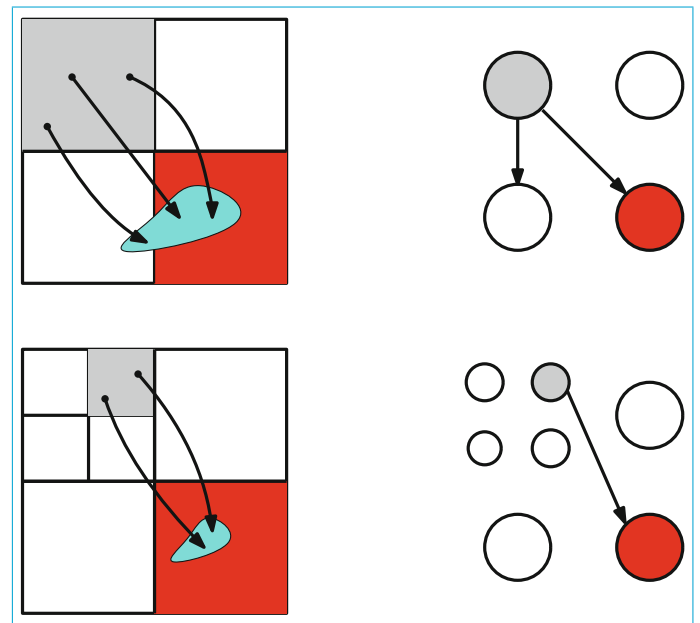


Figure 8 – Haut : l'espace d'état est décomposé en quatre régions, représentées par quatre états discrets. L'ensemble atteignable à partir de la région grise n'est pas garanti d'atteindre la région rouge. Bas : pour améliorer la précision, la région grise est raffinée à travers la redécomposition de la région en quatre sous-régions. Dans ce cas, l'ensemble atteignable à partir de la nouvelle région grise est garanti d'atteindre la région rouge

4.1.3 Approches sans discrétisation de l'espace d'état

Dans [23] et [24], il a été démontré que l'on peut réaliser l'abstraction symbolique de systèmes de contrôle non linéaires incrémentalement stables sans recourir à une discrétisation de l'espace d'état. Cette méthode d'abstraction permet également de préserver une notion d'équivalence approchée (appelée « **bisimilarité approximative** » [1]) entre le modèle symbolique et le système original. La technique utilise une longueur fixe de séquence de commandes quantifiées en tant qu'état du modèle symbolique, contribuant ainsi à atténuer le fléau de la dimensionnalité. La longueur des séquences de commandes, appelée « horizon temporel », est utilisée comme paramètre pour ajuster la précision du modèle symbolique : une valeur plus élevée de l'horizon implique un modèle symbolique plus précis, mais également avec un plus grand nombre d'états symboliques.

4.1.4 Approches compositionnelles

Pour les grands systèmes constitués de plusieurs sous-systèmes interconnectés, une façon d'atténuer le fléau de dimensionnalité est de développer des approches compositionnelles pour utiliser les techniques de contrôle symbolique localement sur chaque sous-système. Ce type d'approche se base sur le principe de « diviser pour régner » et consiste à décomposer un problème global en des sous-problèmes qui peuvent être résolus en utilisant les techniques existantes. Deux types de méthodes compositionnelles ont été proposés dans la littérature :

■ Abstraction compositionnelle

Étant donné un système constitué de plusieurs sous-systèmes et une spécification globale, on commence par appliquer l'étape d'abstraction symbolique localement pour chacun des sous-systèmes. Ensuite, un modèle symbolique global pour le système interconnecté est obtenu en composant les modèles symboliques locaux des différents sous-systèmes. Finalement, la synthèse de contrôleur est réalisée de façon monolithique comme dans la section 2, sur le modèle symbolique global ainsi obtenu. Pour assurer que la composition des modèles symboliques locaux représente bien une abstraction du système interconnecté, plusieurs outils ont été développés dans la littérature, tels que l'utilisation du théorème de petit gain [25], les propriétés de passivité [26] et la notion de composition approchée [27]. La procédure du calcul compositionnel d'abstractions symboliques est illustrée dans la figure 9a.

■ Synthèse compositionnelle de contrôleur

Étant donné un système constitué de plusieurs composantes et une spécification globale, on commence par décomposer la spécification globale en des contrats entre les sous-systèmes. Ainsi, chaque sous-système se voit attribuer un contrat [28] [29], défini à base de « d'hypothèses » et de « garanties », et spécifiant les garanties qu'un sous-système doit satisfaire lorsque le comportement des autres sous-systèmes est cohérent avec les hypothèses. Les contrats doivent être soigneusement choisis de manière à garantir que la satisfaction de tous les contrats par les sous-systèmes assurera la satisfaction de la spécification globale par le système interconnecté. Pour chaque sous-système et son contrat associé, on calcule un modèle symbolique local en se basant sur l'approche présentée dans la section 1. Ensuite, la synthèse de contrôleur est également réalisée localement de façon à assurer la satisfaction du contrat. Dans ce cas, la méthode de synthèse présentée dans la section 2 doit être adaptée, puisque la spécification est modélisée comme un contrat et non comme un ensemble de trajectoires pour le système. Différentes approches ont été proposées dans la littérature pour faire de la synthèse compositionnelle de contrôleurs symboliques pour différentes classes de systèmes et de spécifications [30] [31]. La procédure du calcul compositionnel d'abstractions symboliques est illustrée dans la figure 9b.

4.2 Abstractions complètes

Comme évoqué dans la section 1.2, un contrôleur satisfaisant une propriété sur un modèle symbolique est garanti de satisfaire cette même propriété sur le système de départ, du fait de la relation comportementale décrite dans le théorème 1. La grande majorité des approches existantes d'abstraction symbolique souffrent de problèmes de **conservatisme** dans le sens où certaines trajectoires des modèles symboliques ne sont pas des trajectoires du système d'origine. Par conséquent, si nous ne trouvons pas de contrôleur pour le modèle symbolique, nous ne pouvons pas nécessairement en conclure qu'il n'existe pas de contrôleur pour le système d'origine.

Ceci est en opposition avec les méthodes d'abstractions **complètes**, pour lesquelles l'existence d'un contrôleur pour le modèle symbolique est équivalente à l'existence d'un contrôleur pour le système d'origine. Le problème de construction d'abstractions symboliques complètes reste, en général, ouvert. L'une des rares classes de systèmes pour lesquelles il existe aujourd'hui des méthodes d'abstraction complète est la classe des systèmes incrémentalement stables [32] [33]. Il convient toutefois de souligner que la condition de stabilité incrémentale [34] est particulièrement contraignante, surpassant même les exigences du concept classique de stabilité dans l'automatique. Une autre approche est apparue récemment dans [35], où une abstraction complète peut être réalisée pour une version perturbée du système original.

4.3 Abstractions basées sur les données

Les techniques d'abstraction classiques (comme celle présentée dans cet article) reposent généralement sur un modèle mathématique du système, tel que les modèles basés sur les principes de la physique. Cependant, un modèle analytique peut ne pas être disponible lorsqu'il s'agit de systèmes complexes constitués de composants de natures différentes, décrits par des équations différentielles, des tables de recherche, des systèmes de transitions et des automates hybrides. Dans ce cas, une pratique courante consiste à utiliser des approches basées sur les données. Dans ce type d'approche, la dynamique du système f donnée dans l'équation (1) est inconnue, et on a juste accès à un ensemble de données $\mathcal{D} = \{(x_i, u_i, x_i^+) \in \mathbb{X} \times \mathbb{U} \times \mathbb{X}\}_{i=1, \dots, N'}$ où $(x_i, u_i, x_i^+) \in \mathcal{D}$ implique qu'il existe $w \in \mathbb{W}$ tel que $x_i^+ = f(x_i, u_i, w)$:

Deux types d'approche d'abstraction symbolique se basant sur des données existent actuellement dans la littérature :

■ Une première approche consiste à apprendre un modèle (ou une surapproximation du modèle), puis à abstraire le modèle obtenu par un modèle symbolique comme présenté dans la section 1. Cette approche peut s'appuyer sur des propriétés structurelles du modèle en question, telles que la monotonie [36], afin d'obtenir des garanties déterministes sur le contrôleur obtenu. Alternativement, cette approche peut faire l'hypothèse que le modèle peut être généré à partir d'un espace de Hilbert à noyau reproduisant, afin d'obtenir des garanties déterministes [37] ou probabilistes [38].

■ Une deuxième approche consiste à apprendre directement un modèle symbolique à partir des données. Vu que ce type d'approche ne se base que sur des données sans utiliser des hypothèses sur les propriétés structurelles du système, ce type d'approche ne permet en général d'obtenir que des garanties probabilistes. La question importante dans ce genre d'approche est de déterminer le nombre de données nécessaires à utiliser pour assurer une certaine probabilité de garantie sur le contrôleur construit pour le modèle symbolique. L'inconvénient majeur de ce type d'approche réside dans le fait que la garantie probabiliste décroît avec le temps (c'est-à-dire que plus le domaine temporel sur lequel la trajectoire est définie est long, plus la probabilité décroît), ce qui limite l'applicabilité de ce type d'approche

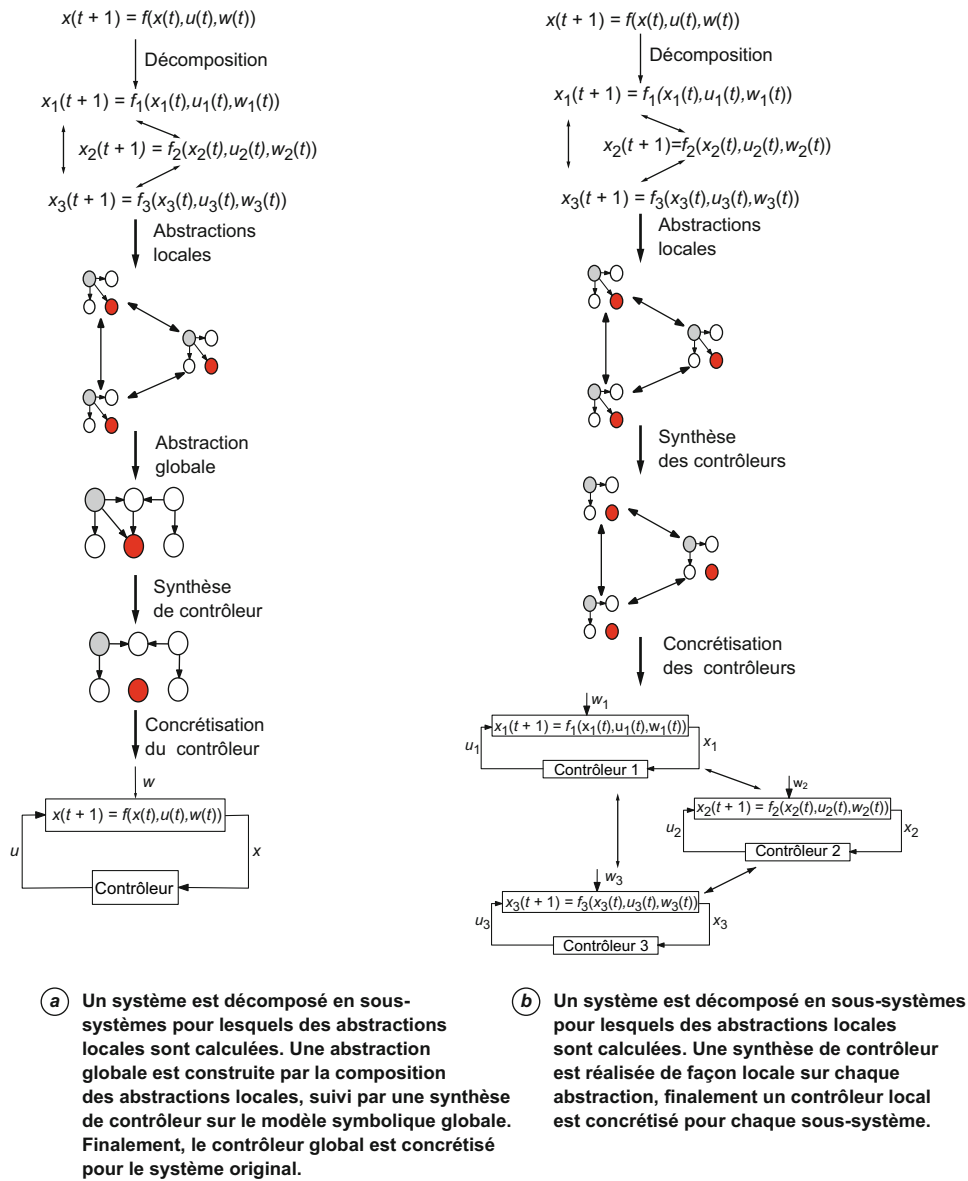


Figure 9 – Deux types d'approche compositionnelle

pour des spécifications à horizon infini, telles que la propriété de sûreté discutée dans la section 2.1. Ce type d'approche a fait l'objet de plusieurs contributions récemment [39] [40] [41] [42].

4.4 Applications du contrôle symbolique

Les techniques de contrôle symbolique ont été appliquées à une variété d'études de cas dans divers domaines. Parmi les exemples

d'application, citons les systèmes de régulation de température [43], le contrôle des véhicules autonomes [44] [45], les modèles de biologie moléculaire [46], le contrôle du glucose pour le diabète [47], le contrôle du flux de trafic [48], la gestion des réseaux électriques [49] [50], les bâtiments actifs [51], et les dynamiques de populations [52]. Des études expérimentales sur les techniques de contrôle symbolique ont été réalisées dans [53] pour les drones, dans [54] [55] pour les robots mobiles, dans [56] pour les systèmes de régulation de température, et dans [57] pour le contrôle des robots bipèdes.

À retenir

- Les techniques de contrôle symbolique présentent l’avantage de pouvoir traiter de façon automatique, formelle et robuste une large classe de systèmes et de spécifications.
- L’étape de discrétisation lors de la construction de modèles symboliques rend ces approches difficiles à appliquer pour des systèmes de grande dimension. Des progrès sont faits dans ce sens en utilisant des outils tels que les abstractions multiéchelles ou les approches compositionnelles.
- La classe des systèmes pour lesquels on peut appliquer des méthodes d’abstraction complète est très limitée. La question de l’abstraction complète d’un système non linéaire est un problème ouvert.
- Motivé par les récentes avancées technologiques dans les domaines de collecte et traitement de données à des niveaux de détails élevés et à grande échelle, la question de construction d’abstraction symbolique à partir des données attire de plus en plus d’attention dans la communauté. Le défi principal reste l’obtention de garanties en se basant uniquement sur des données.

5. Conclusion

Les approches de contrôle symbolique présentent de nombreux avantages :

- une grande généralité sur les systèmes et objectifs de contrôle complexes pouvant être traités ;
- une structure de synthèse de contrôleur complètement automatique et indépendante de la complexité, des incertitudes et des contraintes du système considéré ;
- une robustesse intrinsèque du fait de l’abstraction du système par une surapproximation de l’ensemble de ses comportements ;
- et des garanties formelles de la satisfaction des spécifications par le système original bien que le contrôleur ait été synthétisé sur son modèle symbolique.

La combinaison de tous ces éléments rend les approches à base d’abstraction symbolique très intéressantes dans de nombreux domaines, et la généralisation de leur utilisation va très vraisemblablement se poursuivre dans les années à venir. Bien que cet article se soit focalisé sur le cas de systèmes à dynamiques déterministes, il est à noter qu’il existe également de nombreux travaux sur les abstractions symboliques de systèmes stochastiques [58].

Bien que le passage à l’échelle pour le traitement de systèmes de grande dimension soit l’un des principaux défis des méthodes symboliques, la constante amélioration des puissances de calcul atténue progressivement cette limitation. En particulier, l’introduction d’outils d’accélération se basant sur la parallélisation des calculs réalisés sur des cartes graphiques [59] a le potentiel de grandement faciliter l’application de méthodes symboliques à des systèmes de plus grande taille.

Un autre point d’évolution importante est celui lié aux méthodes basées sur les données qui ont récemment vu le jour dans le domaine des approches d’abstraction symbolique. En effet, la disponibilité toujours grandissante des quantités de données collectées, combinée à l’augmentation des puissances de calcul, va rendre ce type d’approche de plus en plus viable dans tous les domaines, y compris celui des méthodes symboliques. Un point d’attention doit tout de même être évoqué quant à l’utilisation de méthodes exclusivement basées sur des données, sans hypothèse et raisonnement mathématique, qui présente le risque de perdre les garanties formelles et déterministes actuellement associées

aux méthodes symboliques au profit de simples résultats probabilistes. Cependant, il est possible que les quantités de données et puissances de calcul disponibles atteignent éventuellement un niveau permettant à des garanties probabilistes d’être suffisamment hautes pour que le comportement du système puisse tout de même être certifié conforme aux normes en vigueur.

6. Glossaire

Abstraction symbolique ; *symbolic abstraction*

Procédure consistant à approximer un modèle initial par un autre modèle (dit « symbolique ») dont les états et comportements peuvent être vus comme des symboles représentant des ensembles d’états et de comportements du modèle initial.

Analyse d’atteignabilité ; *reachability analysis*

Procédure permettant d’évaluer (généralement via des surapproximations) l’ensemble des successeurs atteignables en un temps fini par un système donné, à partir d’un ensemble d’états initiaux.

Atteignabilité ; *reachability*

Objectif de contrôle dual de la sûreté, où l’on souhaite qu’un système atteigne une région donnée en un temps fini mais indéterminé.

Concrétisation ; *concretisation*

Procédure duale de l’étape d’abstraction symbolique, dans laquelle une loi de commande obtenue pour un modèle symbolique (dans le domaine discret) est adaptée pour être applicable au système initial (dans le domaine continu).

Fonction multivaluée ; *multivalued function*

Fonction pour laquelle une unique valeur d’entrée peut être associée à plusieurs valeurs de sortie.

Modèle symbolique ; *symbolic model*

Modèle mathématique, généralement représenté sous la forme d’un système dynamique avec un nombre fini d’états et de commandes, obtenu à l’issue de l’application d’une procédure d’abstraction symbolique.

Sûreté ; *safety*

Objectif de contrôle où l’on souhaite maintenir en tout temps un système dans une région donnée.

Dans un contexte de gestion des risques, le terme anglais « *safety* » désigne la prévention contre les risques involontaires ou accidentels, alors que « *security* » fait référence à la prévention contre les risques volontaires et malveillants. La traduction française de ces deux termes ne fait malheureusement pas consensus. On peut par exemple entendre parler de sécurité ferroviaire, sécurité civile, et sûreté nucléaire contre les risques involontaires, et de sûreté ferroviaire, sûreté nationale, sécurité nucléaire, et cyber-sécurité contre les risques volontaires. Dans cet article, nous utilisons le terme « sûreté » pour désigner la gestion des risques involontaires et non malveillants, correspondant donc à l’anglais « *safety* ».

Synthèse de contrôleur ; *controller synthesis*

Procédure de calcul ou de création d’une loi de commande permettant que le système en boucle fermée d’un modèle donné satisfasse un objectif de contrôle donné.

7. Sigles, notations et symboles

| Symbole | Description |
|--------------|--|
| \emptyset | Ensemble vide |
| \mathbb{N} | Ensemble des entiers non négatifs $\{0, 1, 2, \dots\}$ |
| \mathbb{R} | Ensemble des réels |

| Symbole | Description |
|------------------------------------|--|
| \mathbb{R}^n | Ensemble des vecteurs réels à n éléments |
| $\text{cl}(\cdot)$ | Adhérence (<i>closure</i> en anglais) d'un ensemble |
| 2^X | Ensemble de tous les sous-ensembles de X |
| $\mathcal{F}(X, Y)$ | Ensemble des fonctions de X vers Y |
| $\frac{\partial f(x)}{\partial x}$ | Dérivée partielle d'une fonction f par rapport à la variable x |

Approches symboliques pour le contrôle des systèmes non linéaires

par **Antoine GIRARD**

Directeur de recherche CNRS
Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes,
91190, Gif-sur-Yvette, France

Pierre-Jean MEYER

Chargé de recherche du développement durable
Univ Gustave Eiffel, COSYS-ESTAS, F-59666 Villeneuve d'Ascq, France

et **Adnane SAOUD**

Professeur Assistant
College of Computing, Mohammed VI Polytechnic University (UM6P), Benguerir, Maroc

Sources bibliographiques

- [1] TABUADA (P.). – *Verification and control of hybrid systems : a symbolic approach*. Springer Science & Business Media (2009).
- [2] REISSIG (G.), WEBER (A.) et RUNGGER (M.). – *Feedback refinement relations for the synthesis of symbolic controllers*. IEEE Transactions on Automatic Control, 62(4) :1781–1796 (2016).
- [3] ALTHOFF (M.), FREHSE (G.) et GIRARD (A.). – *Set propagation techniques for reachability analysis*. Annual Review of Control, Robotics, and Autonomous Systems, 4 :369–395 (2021).
- [4] ANGELI (D.) et SONTAG (E.D.). – *Monotone control systems*. IEEE Transactions on Automatic Control, 48(10) :1684–1698 (2003).
- [5] MEYER (P.-J.), DEVONPORT (A.) et ARCAK (M.). – *Interval Reachability Analysis: Bounding Trajectories of Uncertain Systems with Boxes for Control and Verification*. Springer (2021).
- [6] MEYER (P.-J.), DEVONPORT (A.) et ARCAK (M.). – *TIRA : Toolbox for interval reachability analysis*. ACM International Conference on Hybrid Systems: Computation and Control, p. 224–229 (2019).
- [7] BERTSEKAS (D.). – *Dynamic programming and optimal control: Volume I*. Athena scientific (2012).
- [8] CAMACHO (E.F.) et BORDONS (C.). – *Model Predictive Controllers*, p. 13–30. Springer London London (2007).
- [9] MEYER (P.-J.). – *Invariance and symbolic control of cooperative systems for temperature regulation in intelligent buildings*. PhD thesis Université Grenoble Alpes (2015).
- [10] AZAKI (Z.), GIRARD (A.) et OLARU (S.). – *Predictive and symbolic control: Performance and safety for non-linear systems*. IFAC PapersOnLine, 55(16) : 290–295 (2022).
- [11] SIPSER (M.). – *Introduction to the theory of computation*. ACM Sigact News, 27(1) :27–29 (1996).
- [12] BAIER (C.) et KATOEN (J.-P.). – *Principles of model checking*. MIT press (2008).
- [13] BELTA (C.), YORDANOV (B.) et GOL (E.A.). – *Formal methods for discrete-time dynamical systems*. Springer (2017).
- [14] PLATZER (A.) et QUESEL (J.-D.). – *Keymaera: A hybrid theorem prover for hybrid systems (system description)*. International Joint Conference on Automated Reasoning, p. 171–178 (2008).
- [15] LIU (J.) et OZAY (N.). – *Finite abstractions with robustness margins for temporal logic-based control synthesis*. Nonlinear Analysis: Hybrid Systems, 22 :1–15 (2016).
- [16] APAZA-PEREZ (W.A.), GIRARD (A.), COMBASTEL (C.) et ZOLGHADRI (A.). – *Symbolic observer-based controller for uncertain nonlinear systems*. IEEE Control Systems Letters, 5(4) :1297–1302 (2020).
- [17] GIRARD (A.), GÖSSLER (G.) et MOUELHI (S.). – *Safety controller synthesis for incrementally stable switched systems using multiscale symbolic models*. IEEE Transactions on Automatic Control, 61(6) :1537–1549 (2015).
- [18] HSU (K.), MAJUMDAR (R.), MALLIK (K.) et SCHMUCK (A.-K.). – *Multi-layered abstraction-based controller synthesis for continuous-time systems*. ACM International Conference on Hybrid Systems: Computation and Control, p. 120–129 (2018).
- [19] HUSSEIN (O.) et TABUADA (P.). – *Lazy controller synthesis using three-valued abstractions for safety and reachability specifications*. IEEE Conference on Decision and Control, p. 3567–3572 (2018).
- [20] IVANOVA (E.), SAOUD (A.) et GIRARD (A.). – *Lazy controller synthesis for monotone transition systems and directed safety specifications*. Automatica, 135 :109993 (2022).
- [21] SIBAI (H.) et MITRA (S.). – *Symmetry-based abstractions for hybrid automata*. IEEE Transactions on Automatic Control, In press (2024).
- [22] HSU (K.), MAJUMDAR (R.), MALLIK (K.) et SCHMUCK (A.-K.). – *Lazy abstraction-based control for safety specifications*. IEEE Conference on Decision and Control, p. 4902–4907 (2018).
- [23] LE CORRONC (E.), GIRARD (A.) et GOESSLER (G.). – *Mode sequences as symbolic states in abstractions of incrementally stable switched systems*. IEEE Conference on Decision and Control, p. 3225–3230 (2013).
- [24] GIRARD (A.). – *Approximately bisimilar abstractions of incrementally stable finite or infinite dimensional systems*. IEEE Conference on Decision and Control, p. 824–829 (2014).
- [25] RUNGGER (M.) et ZAMANI (M.). – *Compositional construction of approximate abstractions of interconnected control systems*. IEEE Transactions on Control of Network Systems, 5(1) :116–127 (2016).
- [26] ZAMANI (M.) et ARCAK (M.). – *Compositional abstraction for networks of control systems: A dissipativity approach*. IEEE Transactions on Control of Network Systems, 5(3) :1003–1015 (2017).
- [27] SAOUD (A.), JAGTAP (P.), ZAMANI (M.) et GIRARD (A.). – *Compositional abstraction-based synthesis for interconnected systems: An approximate composition approach*. IEEE

- Transactions on Control of Network Systems, 8(2) :702–712 (2021).
- [28] BENVENISTE (A.), CAILLAUD (B.), NICKOVIC (D.), PASSERONE (R.), RACLET (J.-B.), REINKEMEIER (P.), SANGIOVANNI-VINCENTELLI (A.), DAMM (W.), HENZINGER (T.A.) et LARSEN (K.G.). – *Contracts for system design*. Foundations and Trends® in Electronic Design Automation, 12(2-3) :124–400 (2018).
- [29] SAOUD (A.), GIRARD (A.) et FRIBOURG (L.). – *Assume-guarantee contracts for continuous-time systems*. Automatica, 134 :109910 (2021).
- [30] MEYER (P.-J.) et DIMAROGONAS (D.V.). – *Compositional abstraction refinement for control synthesis*. Nonlinear Analysis: Hybrid Systems, 27 :437–451 (2018).
- [31] SAOUD (A.), GIRARD (A.) et FRIBOURG (L.). – *Contract-based design of symbolic controllers for safety in distributed multiperiodic sampled-data systems*. IEEE Transactions on Automatic Control, 66(3) :1055–1070 (2020).
- [32] POLA (G.), GIRARD (A.) et TABUADA (P.). – *Approximately bisimilar symbolic models for nonlinear control systems*. Automatica, 44(10) :2508–2516 (2008).
- [33] GIRARD (A.), POLA (G.) et TABUADA (P.). – *Approximately bisimilar symbolic models for incrementally stable switched systems*. IEEE Transactions on Automatic Control, 55(1) :116–126 (2010).
- [34] ANGELI (D.). – *A Lyapunov approach to incremental stability properties*. IEEE Transactions on Automatic Control, 47(3) :410–421 (2002).
- [35] LIU (J.). – *Robust abstractions for control synthesis: Completeness via robustness for linear-time properties*. ACM International Conference on Hybrid Systems: Computation and Control, p. 101–110 (2017).
- [36] MAKDESI (A.), GIRARD (A.) et FRIBOURG (L.). – *Data-driven models of monotone systems*. IEEE Transactions on Automatic Control, In press (2024).
- [37] HASHIMOTO (K.), SAOUD (A.), KISHIDA (M.), USHIO (T.) et DIMAROGONAS (D.V.). – *Learning-based symbolic abstractions for nonlinear control systems*. Automatica, 146 :110646 (2022).
- [38] JACKSON (J.), LAURENTI (L.), FREW (E.) et LAHIJANIAN (M.). – *Safety verification of unknown dynamical systems via gaussian process regression*. IEEE Conference on Decision and Control, p. 860–866 (2020).
- [39] DEVONPORT (A.), SAOUD (A.) et ARCAK (M.). – *Symbolic abstractions from data: A PAC learning approach*. IEEE Conference on Decision and Control, p. 599–604 (2021).
- [40] LAVAEI (A.), SOUDJANI (S.), FRAZZOLI (E.) et ZAMANI (M.). – *Constructing mdp abstractions using data with formal guarantees*. IEEE Control Systems Letters, 7 :460–465 (2022).
- [41] BANSE (A.), ROMAO (L.), ABATE (A.) et JUNGERS (R.M.). – *Data-driven abstractions via adaptive refinements and a kantorovich metric* [extended version]. arXiv preprint, arXiv :2303.17618 (2023).
- [42] BANSE (A.), ROMAO (L.), ABATE (A.) et JUNGERS (R.). – *Data-driven memory-dependent abstractions of dynamical systems*. Learning for Dynamics and Control Conference, p. 891–902 (2023).
- [43] POLA (G.), PEPE (P.) et DI BENEDETTO (M.D.). – *Decentralized supervisory control of networks of nonlinear control systems*. IEEE Transactions on Automatic Control, 63(9) :2803–2817 (2017).
- [44] KHALED (M.), KIM (E.S.), ARCAK (M.) et ZAMANI (M.). – *Synthesis of symbolic controllers: A parallelized and sparsity-aware approach*. International Conference on Tools and Algorithms for the Construction and Analysis of Systems, p. 265–281 (2019).
- [45] SMITH (S.W.), SAOUD (A.) et ARCAK (M.). – *Monotonicity-based symbolic control for safety in driving scenarios*. IEEE Control Systems Letters, 6 :830–835 (2021).
- [46] BANERJEE (T.), MAJUMDAR (R.), MALLIK (K.), SCHMUCK (A.-K.) et SOUDJANI (S.). – *A direct symbolic algorithm for solving stochastic Rabin games*. International Conference on Tools and Algorithms for the Construction and Analysis of Systems, p. 81–98 (2022).
- [47] BORRI (A.), POLA (G.), PEPE (P.), DI BENEDETTO (M.D.) et PALUMBO (P.). – *Symbolic control design of an artificial pancreas for type-2 diabetes*. IEEE Transactions on Control Systems Technology, 30(5) :2131–2146 (2021).
- [48] COOGAN (S.), ARCAK (M.) et BELTA (C.). – *Formal methods for control of traffic flow: Automated control synthesis from finite-state transition models*. IEEE Control Systems Magazine, 37(2) :109–128 (2017).
- [49] ZONETTI (D.), SAOUD (A.), GIRARD (A.) et FRIBOURG (L.). – *A symbolic approach to voltage stability and power sharing in time-varying dc microgrids*. European Control Conference, p. 903–909 (2019).
- [50] WOODING (B.), VAHIDINASAB (V.) et SOUDJANI (S.). – *Formal controller synthesis for frequency regulation utilising electric vehicles*. International Conference on Smart Energy Systems and Technologies, p. 1–6 (2020).
- [51] WOODING (B.), VAHIDINASAB (V.) et SOUDJANI (S.). – *Operation and control of a population of active buildings at network level*. Active Building Energy Systems : Operation and Control, Springer, p. 193–218 (2021).
- [52] COOGAN (S.) et ARCAK (M.). – *Finite abstraction of mixed monotone systems with discrete and continuous inputs*. Nonlinear Analysis : Hybrid Systems, 23 :254–271 (2017).
- [53] KREUZER (M.), WEBER (A.), LEUPOLZ (C.) et KNOLL (A.). – *Symbolic control applied to miniature quadcopter mission guidance*. European Control Conference, p. 1–8 (2023).
- [54] SUNDARSINGH (D.S.), SAOUD (A.) et JAGTAP (P.). – *Barrier function-based distributed symbolic controller for multi-agent systems*. arXiv 2304.01652 (2023).
- [55] SUNDARSINGH (D.S.), BHAGIYA (J.), CHATROLA (J.) et JAGTAP (P.). – *Autonomous exploration using ground robots with safety guarantees*. IEEE/RSJ International Conference on Intelligent Robots and Systems, p. 9745–9750 (2023).
- [56] MEYER (P.-J.), GIRARD (A.) et WITRANT (E.). – *Symbolic control of monotone systems application to ventilation regulation in buildings*. International Conference on Hybrid Systems : Computation and Control, p. 281–282 (2015).
- [57] AMES (A.D.), TABUADA (P.), JONES (A.), MA (W.-L.), RUNGGER (M.), SCHÜRMANN (B.), KOLATHAYA (S.) et GRIZZLE (J.W.). – *First steps toward formal controller synthesis for bipedal robots with experimental implementation*. Nonlinear Analysis : Hybrid Systems, 25 :155–173 (2017).
- [58] LAVAEI (A.), SOUDJANI (S.), ABATE (A.) et ZAMANI (M.). – *Automated verification and synthesis of stochastic hybrid systems: A survey*. Automatica, 146 :110617 (2022).
- [59] KHALED (M.) et ZAMANI (M.). – *pFaces : An acceleration ecosystem for symbolic control*. ACM International Conference on Hybrid Systems : Computation and Control, p. 252–257 (2019).

À lire également dans nos bases

QUEINNEC (I.), TARBOURIECH (S.) et ZACCARIAN (L.). – *Tour d'horizon sur les systèmes dynamiques hybrides*. [S 7 434] (2023).

ENCRENAZ-TIPHENE (E.). – *Méthodes formelles pour la vérification des systèmes embarqués*. [H 8 250] (2013).

Outils logiciels

SCOTS : A Tool for the Synthesis of Symbolic Controllers
<https://gitlab.lrz.de/hcs/scots>

pFaces : An Acceleration Ecosystem for Formal Methods in Control
<https://github.com/parallall/pFaces>

Co4Pro : Correct by Construction Controllers from Control Programs
<https://github.com/girardan/Co4Pro>

RObustly Complete control Synthesis (ROCS)
<https://git.uwaterloo.ca/hybrid-systems-lab/rocs>

QUEST : A Tool for State-Space Quantization-Free Synthesis of Symbolic Controllers

<https://gitlab.lrz.de/hcs/QUEST>

CoSyMA : A Tool for Controller Synthesis Using Multi-scale Abstractions

<https://github.com/mouelhis/cosymast>

TIRA : Toolbox for Interval Reachability Analysis

https://gitlab.com/pj_meyer/TIRA

Continuous Reachability Analyzer (CORA)

<https://github.com/TUMcps/CORA>

Sites internet

IEEE Technical Committee on Hybrid Systems

<https://ieeecss.org/tc/hybrid-systems>

IFAC Technical Committee on Discrete Event Systems and Hybrid Systems

<https://tc.ifac-control.org/1/3>

The IFAC journal on Nonlinear Analysis : Hybrid Systems (NAHS)

<https://www.sciencedirect.com/journal/nonlinear-analysis-hybrid-systems>

Événements

ACM International Conference on Hybrid Systems : Computation and Control.

A lieu tous les ans.

<https://hscs.acm.org>

IFAC Conference on Analysis and Design of Hybrid Systems. A lieu tous les trois ans.

<https://tc.ifac-control.org/1/3/adhs-series>

Brevets

Mahmoud Khaled and Majid Zamani. Distributed Automated Synthesis Of Correct-by-Construction Controllers. European Patent Office, EP3633468A1, WO2020070206A1, 2020.

<https://worldwide.espacenet.com/patent/search/family/064017240/publication/EP3633468A1?q=pn%3DEP3633468A1>

Annuaire

Documentation – Formation – Séminaires (liste non exhaustive)

Antoine Girard, Symbolic control of nonlinear systems : safety, optimization and learning Cours, École d'Été d'Automatique de Grenoble, 2023

<https://hal.science/hal-04410084>

Séminaire, Autonomy Talks, ETH Zürich

<https://www.youtube.com/watch?v=be9mE8Kp9Jc>

Gagnez du temps et sécurisez vos projets en utilisant une source actualisée et fiable



RÉDIGÉE ET VALIDÉE
PAR DES EXPERTS




MISE À JOUR
PERMANENTE



100 % COMPATIBLE
SUR TOUS SUPPORTS
NUMÉRIQUES



SERVICES INCLUS
DANS CHAQUE OFFRE

- + de 340 000 utilisateurs chaque mois
- + de 10 000 articles de référence et fiches pratiques
- Des Quiz interactifs pour valider la compréhension 

SERVICES ET OUTILS PRATIQUES



Questions aux experts*

Les meilleurs experts techniques et scientifiques vous répondent



Articles Découverte

La possibilité de consulter des articles en dehors de votre offre



Dictionnaire technique multilingue

45 000 termes en français, anglais, espagnol et allemand



Archives

Technologies anciennes et versions antérieures des articles



Info parution

Recevez par email toutes les nouveautés de vos ressources documentaires

*Questions aux experts est un service réservé aux entreprises, non proposé dans les offres écoles, universités ou pour tout autre organisme de formation.

Les offres Techniques de l'Ingénieur



INNOVATION

- Éco-conception et innovation responsable
- Nanosciences et nanotechnologies
- Innovations technologiques
- Management et ingénierie de l'innovation
- Smart city – Ville intelligente



MATÉRIAUX

- Bois et papiers
- Verres et céramiques
- Textiles
- Corrosion – Vieillessement
- Études et propriétés des métaux
- Mise en forme des métaux et fonderie
- Matériaux fonctionnels. Matériaux biosourcés
- Traitements des métaux
- Élaboration et recyclage des métaux
- Plastiques et composites



MÉCANIQUE

- Frottement, usure et lubrification
- Fonctions et composants mécaniques
- Travail des matériaux – Assemblage
- Machines hydrauliques, aérodynamiques et thermiques
- Fabrication additive – Impression 3D



ENVIRONNEMENT – SÉCURITÉ

- Sécurité et gestion des risques
- Environnement
- Génie écologique
- Technologies de l'eau
- Bruit et vibrations
- Métier : Responsable risque chimique
- Métier : Responsable environnement



ÉNERGIES

- Hydrogène
- Ressources énergétiques et stockage
- Froid industriel
- Physique énergétique
- Thermique industrielle
- Génie nucléaire
- Conversion de l'énergie électrique
- Réseaux électriques et applications



GÉNIE INDUSTRIEL

- Industrie du futur
- Management industriel
- Conception et production
- Logistique
- Métier : Responsable qualité
- Emballages
- Maintenance
- Traçabilité
- Métier : Responsable bureau d'étude / conception



ÉLECTRONIQUE – PHOTONIQUE

- Electronique
- Technologies radars et applications
- Optique – Photonique



TECHNOLOGIES DE L'INFORMATION

- Sécurité des systèmes d'information
- Réseaux Télécommunications
- Le traitement du signal et ses applications
- Technologies logicielles – Architectures des systèmes
- Sécurité des systèmes d'information



AUTOMATIQUE – ROBOTIQUE

- Automatique et ingénierie système
- Robotique



INGÉNIERIE DES TRANSPORTS

- Véhicule et mobilité du futur
- Systèmes aéronautiques et spatiaux
- Systèmes ferroviaires
- Transport fluvial et maritime



MESURES – ANALYSES

- Instrumentation et méthodes de mesure
- Mesures et tests électroniques
- Mesures mécaniques et dimensionnelles
- Qualité et sécurité au laboratoire
- Mesures physiques
- Techniques d'analyse
- Contrôle non destructif



PROCÉDÉS CHIMIE – BIO – AGRO

- Formulation
- Bioprocédés et bioproductions
- Chimie verte
- Opérations unitaires. Génie de la réaction chimique
- Agroalimentaire



SCIENCES FONDAMENTALES

- Mathématiques
- Physique Chimie
- Constantes physico-chimiques
- Caractérisation et propriétés de la matière



BIOMÉDICAL – PHARMA

- Technologies biomédicales
- Médicaments et produits pharmaceutiques



CONSTRUCTION ET TRAVAUX PUBLICS

- Droit et organisation générale de la construction
- La construction responsable
- Les superstructures du bâtiment
- Le second œuvre et l'équipement du bâtiment
- Vieillessement, pathologies et réhabilitation du bâtiment
- Travaux publics et infrastructures
- Mécanique des sols et géotechnique
- Préparer la construction
- L'enveloppe du bâtiment
- Le second œuvre et les lots techniques