Algorithme de résolution de contraintes intervalles pour la vérification de systèmes dynamiques continus

Stage Master 2, 2018–2019

Thème : Analyse intervalle, équations différentielles, logique du premier ordre.

Laboratoire : U2IS, ENSTA ParisTech

Adresse : 828 boulevard des maréchaux 91762 Palaiseau Cedex

Encadrants : Alexandre Chapoutot et Michel Kieffer
`alexandre.chapoutot@ensta-paristech.fr`
`michel.kieffer@lss.supelec.fr`

Durée : 5 à 6 mois

Rémunération : suivant la législation en vigueur.

Contexte. Les systèmes de contôle-commande sont des systèmes formés de composants logiciels s'exécutant dans un environnement physique tel qu'il existe une interaction forte entre l'environnement physique et le logiciel. Nous parlons alors de *systèmes hybrides*. Un exemple de tel système est le régulateur de vitesses d'une voiture. Cette interaction met en relation deux mondes qui évoluent différemment dans le temps. Le premier est celui de l'environnement physique qui évolue continûment dans le temps. Le second est celui des programmes informatiques qui évoluent de manière cadencée dans le temps. La validation de tels systèmes est une étape clé lors du processus de développement car un dysfonctionnements pourrait causer de graves dommages. La méthode la plus utilisée est la simulation numérique de modèles mathématiques représentant ces systèmes hybrides, comme ceux de l'outil Matlab/Simulink. Or ces techniques n'offrent aucune garantie sur le comportement globale du système. D'où l'intérêt de l'utilisation de méthodes formelles qui permettent d'accroître la confiance dans le fonctionnement du logiciel en apportant une preuve mathématique de son comportement.

L'une de ces méthodes est la recherche d'invariants, c'est-à-dire une propriété vraie pour toutes les exécutions possibles du logiciel. Un invariant dans le cas des systèmes hybrides est un sous-espace où les comportements du systèmes vivent. Une approche utilisée pour la vérification des propriétés de sûreté, c'est-à-dire montrer que quelque chose de mal ne peut pas arriver, pour les systèmes hybrides s'appuie sur le calcul d'invariants. Un travail récent sur ce sujet est [3] qui définit la notion de *certificats de barrière*. Une barrière est une fonction qui, si elle existe, montre qu'aucun comportement d'un système hybride ne peut atteindre une région considérée comme dangereuse à partir d'un ensemble de départ donné.

Travail à réaliser. Le but du stage est d'étendre une méthode de recherche de certificats de barrière. Une plateforme déjà existante [1] permet de trouver des barrières pour les systèmes purement continue à l'aide des outils de l'analyse par intervalles [2]. Par contre une difficulté de cette approche est de difficilement prouvé qu'une barrière n'existe pas. Une façon de faire est de trouver un contre-exemple. La personne en stage devra développer une telle méthode en s'appuyant sur la formalisation définie dans [4]. Les travaux associés à ce stage mélangent logique du premier ordre et les algorithmes de résolution branch-and-bound.

Profile et candidature. Ce sujet s'adresse à des étudiants en informatique aimant programmer (en C++) et ayant des connaissances en logique. Des connaissances sur l'analyse par intervalles est un plus mais pas obligatoire.

Le candidat devra soumettre par courrier électronique les documents suivants :
— une lettre de motivation ;
— une curriculum vitæ ;
— une copies des diplômes et des relevés de notes de licence et master.

Références

[1] A. Djaballah, A. Chapoutot, M. Kieffer, and O. Bouissou. Construction of parametric barrier functions for dynamical systems using interval analysis. *Automatica*, 78 :287 – 296, Feb. 2017.

[2] L. Jaulin and É. Walter. Guaranteed tuning, with application to robust control and motion planning. *Automatica*, 32(8) :1217–1221, 1996.

[3] S. Prajna and A. Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *Hybrid Systems : Computation and Control*, pages 477–492. Springer, 2004.

[4] S. Prajna and A. Rantzer. Primal–dual tests for safety and reachability. In *Hybrid Systems : Control and Computation*, volume 3414 of *LNCS*, pages 542–556. Springer-Verlag, 2005.