Extension d'un langage dédié à la résolution de problèmes de contrôle optimal

Responsables.

— Olivier Cots, [1] enseignant-chercheur à l'INP-ENSEEIHT de Toulouse ;
— Guillaume Dupont, [2] enseignant-chercheur à l'INP-ENSEEIHT de Toulouse .

Entreprise / Laboratoire. INP-ENSEEIHT-IRIT (UMR CNRS 5505)

Mots-clés. Domain-Specific Language, Contrôle Optimal, Julia.

Contexte. Depuis début 2023, des chercheurs du laboratoire IRIT de Toulouse et des collaborateurs du Centre Inria Université Côte d'Azur développent un ensemble de packages Julia pour le contrôle optimal. Ces packages sont regroupés sous le nom de `control-toolbox` et sont disponibles à l'adresse suivante : https://github.com/control-toolbox. Ces packages ont été présentés à la conférence JuliaCon 2023 au MIT à Cambridge aux États-Unis, aux journées JuliaOpt 2023 à Paris et à la conférence JuliaCon 2024 à Eindhoven. Un des retours les plus positifs de ces conférences est sur la simplicité d'écriture des problèmes de contrôle optimal. La modélisation étant proche de la formulation mathématique, les utilisateurs peuvent rapidement écrire et résoudre des problèmes de contrôle optimal. Voici un exemple de code :

```
using OptimalControl

ocp = @def begin
    t ∈ [0, 1], time
    x ∈ R², state
    u ∈ R, control
    x(0) == [ -1, 0 ]
    x(1) == [ 0, 0 ]
    ẋ(t) == [ x₂(t), u(t) ]
    ∫( 0.5u(t)^2 ) → min
end
```

La macro `@def` se trouve dans le package `CTBase.jl`. [3] D'un point de vue du développeur informatique, le code de cette macro est simplifié par le fait que le langage Julia est un langage qui permet l'introspection et la manipulation de code. De plus, nous avons fait le choix de

1. olivier.cots@toulouse-inp.fr
2. guillaume.dupont@toulouse-inp.fr
3. Disponible à l'adresse https://github.com/control-toolbox/CTBase.jl.

n'autoriser que de la syntaxe pure Julia, autrement dit chaque ligne entre les balises `begin` et `end` est du code que Julia sait parser. Ainsi, l'arbre syntaxique est généré automtiquement. Enfin, des outils puissants permettent de parcourir cet arbre syntaxique pour donner le sens mathématique du code.

Objectifs. Ce projet long vise à étendre et améliorer ce langage dédié (ou Domain-Specific Language ou DSL) à la résolution de problèmes de contrôle optimal. Il sera demandé aux étudiants du projet long de travailler sur les points suivants.

1. Des collaborateurs [4] veulent utiliser notre package afin de définir des problèmes de contrôle optimal qui ne rentrent pas dans la classe des problèmes que nous pouvons modéliser. Nous souhaitons donc ajouter une technique d'annotation afin de permettre à un développeur extérieur de définir son propre problème.
2. Nous ne spécifions pas assez le type des données du problème. Il serait profitable d'utiliser un patron de conception de type Builder [5] pour au final avoir un objet dont tous les types sont spécifiés.
3. Nous voudrions ajouter des possibilités d'écriture plus compactes. Les contraintes de type $u(t) \geq 0$ doivent pour le moment être écrite sur une ligne séparée de la déclaration $u(t) \in \mathbb{R}$. Il serait intéressant de pouvoir écrire $u(t) \in \mathbb{R}_+$. Voir la discussion à l'adresse `https://tinyurl.com/control-toolbox-issue-dsl`.
4. Comme nous pouvons le voir sur le code à la première page, nous devons indiquer explicitement qui sont les variable de temps, de contrôle et d'état. Or, il n'est pas nécessaire de le faire. Mathémtiquement, nous le faisons pas dans la formulation du problème. Le rôle de ces variables est déduit de la formulation du problème.
5. Pour le moment, le problème doit être défini en un seul bloc. Il serait intéressant de pouvoir définir un problème de manière incrémentale.

Compétences / Connaissances minimales requises.

— Des connaissances en langage impératif pour se mettre à Julia sont nécessaires ;
— Des connaissances en traduction des langages sont nécessaires ;
— Aucune connaissance spécifique en théorie du contrôle optimal n'est nécessaire ;

4. Voir `https://github.com/control-toolbox/LossControl.jl`.
5. Voir par exemple `https://refactoring.guru/design-patterns/builder`.