

# Outil d'allocation de contrôle-commande analogique de systèmes de sûreté dans le domaine du nucléaire

Lydie Suh<sup>1</sup>, Xavier Zeitoun<sup>2</sup>,

<sup>1</sup> Framatome Grenoble, France

`lydie.suh@framatome.com`

<sup>2</sup> Université Paris-Saclay, CEA, List, France

`xavier.zeitoun@cea.fr`

**Mots-clés :** *retex, contrôle de centrale nucléaire, allocation, PLNE*

## 1 Introduction

Les systèmes de contrôle des centrales nucléaires sont critiques. Les industriels du domaine proposent plusieurs solutions pour ce contrôle. La société Framatome propose une solution nommée Hardline [1] qui est entièrement analogique, ce qui garantit l'absence de mode commun logiciel. Ces systèmes opèrent sans OS ni langage de programmation, mais directement à partir de carte électroniques reliés entre elles et avec les signaux du procédé.

Si ces systèmes opèrent sans couche numérique, le logiciel est cependant utile et même nécessaire à leur conception. Notamment pour l'allocation des briques élémentaires qui est soumise à plusieurs contraintes. Cette allocation doit également minimiser l'espace pris par le système car fortement contraint dans la zone de contrôle d'une centrale. Les outils de programmation linéaire en nombre entier (PLNE) permettent de résoudre ces problèmes d'allocation.

Au cours du projet collaboratif Orion financé par BPIFrance, le CEA List et Framatome ont développé un prototype d'outil permettant d'accélérer et de rendre plus robuste le travail d'allocation d'un système Hardline.

## 2 Contexte, Problème et outils

### 2.1 Le flux de conception (partiel) d'un système Hardline

Pour la conception d'un système Hardline, le systémier reçoit du donneur d'ordre une première spécification des fonctions de contrôle que le système doit réaliser. A partir de cette spécification, le systémier détermine par quel type de composants d'une solution Hardline (voir section 2.2) sera réalisé cette spécification. On appelle le résultat de cette étape *spécification fonctionnelle*. Il doit ensuite préciser comment ces composants seront alloués. Cette tâche d'allocation, sans assistance outillée est fastidieuse et potentiellement source d'erreurs (452 composants et 1332 connexions sur notre cas d'étude). Cette allocation est ensuite transmise au métier aval qui affinera la conception.

### 2.2 La structure d'un système Hardline

Un système Hardline se présente sous la forme d'un ensemble d'*armoires*. Ces armoires contiennent un ensemble de *châssis*. Chaque châssis contient un ensemble de *modules*. Un module est une carte électronique dédiée à une fonction précise (relais, comparaison à seuil, isolation, conditionnement thermocouple, test ...). Pour optimiser l'espace, chaque module peut héberger plusieurs instances de sa fonction. Par exemple certains modules relais peuvent héberger jusqu'à 12 relais. Les connexions entre le châssis et l'extérieur se font au moyen d'un

*bornier client*. Ce bornier contient des *connecteurs* qui permettent de connecter des signaux venant d'autres châssis, des signaux venant du procédé à contrôler, ou des signaux de test. Au sein d'un même châssis, les connexions entre les modules sont assurés au moyen d'une *carte de configuration* afin de faciliter les opérations de câblage de la solution.

### 2.3 PLNE pour l'allocation de système de contrôle-commande

Définir une allocation d'un système Hardline, c'est définir pour chaque châssis le contenu des modules qui le compose et allouer toutes les connexions qui sortent du châssis sur des connecteurs du bornier client. On crée un problème d'allocation par châssis. Pour cela, le systémier définit une *allocation préliminaire* qui alloue les éléments de la spécification fonctionnelle sur les châssis. Cette étape additionnelle a cependant été favorablement reçue par le systémier car elle lui permet de contrôler d'une certaine "lisibilité" de l'architecture de l'allocation.

Étant donné un ensemble de composants de la spécification fonctionnelle à allouer sur un châssis, on définit un ensemble de modules pour contenir les composants. On sélectionne cet ensemble suffisamment grand pour s'assurer qu'il existe une solution au problème d'optimisation. L'objectif sera de minimiser le nombre de modules réellement utilisés. Ensuite on définit des variables entières pour l'allocation des composants dans les modules et pour l'allocation des connexions vers l'extérieur du châssis sur les connecteurs. Les contraintes linéaires ajoutés au problème, résultent des contraintes de la technologie (exemple : nombre de voies par module, quel type de composant par module ...) et des contraintes d'isolation des signaux au niveau des modules et des connecteurs.

Nous avons formulé une première expression naïve de ce problème d'allocation qui définissait une variable, finalement booléenne, par composant de la spécification fonctionnelle. Les performances en terme de temps de calcul n'étaient pas satisfaisantes. Nous avons donc réalisé une seconde formulation qui tirait parti de la modélisation des variables par des entiers et a permis de résoudre l'allocation en temps raisonnable. Cependant la première formulation reste utile pour une fonctionnalité de mise à jour de l'allocation suite, par exemple, à une modification légère de la spécification fonctionnelle.

L'allocation des châssis dans les armoires est également résolue par PLNE dans une seconde étape en s'assurant que certaines contraintes soient respectées.

### 2.4 Outillage et collaboration

Le prototype résultant de la collaboration permet l'import d'une spécification fonctionnelle, l'édition d'une allocation préliminaire, le calcul d'une allocation du contrôle-commande spécifié, le paramétrage de ce calcul et l'export d'une allocation. L'existence d'un framework d'optimisation open source avec des binding python [2] a grandement facilité l'intégration du module d'optimisation. Le prototype utilise au maximum les outils existants et maîtrisés par le partenaire industriel (notamment Microsoft Visio) ce qui a permis un affinage itératif de la définition du problème d'optimisation. De plus la production de représentations graphiques de la spécification fonctionnelle importée et de l'allocation produite à l'aide de plantuml (<https://plantuml.com/>) a facilité la compréhension mutuelle.

## Références

- [1] Hardline, Framatome, <https://www.framatome.com/solutions-portfolio/portfolio/product?product=A3028&businessUnit=i-c&businessArea=i-c-safety-systems>
- [2] Laurent Perron and Vincent Furnon. OR-Tools, Google, <https://developers.google.com/optimization/>, 2022