

Rapport sur la thèse de doctorat présentée par **Monsieur Pierre FILIOL**

sur le sujet

« **Hardware Acceleration of Interval Arithmetic in the RISC-V and Application to Mobile Robotic** »

en vue de l'obtention du **grade de docteur** en Robotique de l'**ENSTA**

La thèse de Monsieur Pierre Filiol, intitulée « Hardware Acceleration of Interval Arithmetic in RISC-V Architectures and Applications to Mobile Robotics », a été réalisée au Laboratoire en Sciences et Techniques de l'Information, de la Communication et de la Connaissance (Lab-STICC – CNRS), sous la direction de sous la direction de Messieurs Luc Jaulin, Jean-Christophe Le Lann et Théotime Bollengier.

Ce manuscrit, rédigé en anglais, compte 222 pages et s'organise en sept chapitres, auxquels s'ajoutent une introduction générale, des conclusions et perspectives, d'une bibliographie d'articles dont il est fait mention dans la thèse et enfin des annexes. Le document présente les travaux de thèse menés par Monsieur Pierre Filiol, consacrés à l'accélération matérielle de l'arithmétique par intervalles sur architectures RISC-V, dans une perspective d'applications à la robotique mobile embarquée. L'objectif est de lever les limitations de performance qui freinent l'utilisation de ces méthodes sur des systèmes contraints, tout en conservant leurs garanties de correction mathématique.

Après une introduction présentant le contexte scientifique et les enjeux liés à l'arithmétique par intervalles en robotique, le premier chapitre propose un état de l'art des méthodes d'arithmétique par intervalles et des architectures matérielles existantes, en identifiant les verrous limitant leur déploiement sur systèmes embarqués. Le deuxième chapitre introduit les fondements théoriques nécessaires à l'implémentation matérielle de ces méthodes, en précisant les propriétés numériques et les contraintes associées. Le troisième chapitre décrit la conception d'extensions matérielles dédiées à l'arithmétique par intervalles dans le cadre de l'architecture RISC-V. Le quatrième chapitre présente leur intégration au sein d'un processeur et les choix d'architecture retenus pour concilier performance et efficacité énergétique. Le cinquième chapitre propose une évaluation expérimentale des solutions développées, à travers des benchmarks représentatifs. Le sixième chapitre illustre l'apport de ces travaux sur des cas d'usage en robotique mobile. Enfin, le septième chapitre discute les limites des approches proposées et esquisse plusieurs perspectives de recherche.

Le manuscrit est globalement bien structuré et présente une progression cohérente entre aspects théoriques, propositions architecturales et validations expérimentales. Bien que n'étant pas moi-même spécialiste des aspects purement matériels et de la synthèse logique, j'ai trouvé le manuscrit suffisamment pédagogique pour en saisir l'essentiel. Les travaux couvrent à la fois des contributions méthodologiques et des réalisations concrètes, témoignant d'une capacité à articuler des développements matériels avec des applications en robotique.

Ces travaux ont été reconnus par deux publications dans une revue internationale et trois conférences internationales. **En conclusion, au vu des contributions et de la qualité du travail, j'é mets un avis très favorable à la soutenance de la thèse de Monsieur Pierre Filiol.**

Ci-après les commentaires détaillés sur chaque partie du mémoire.

Introduction

L'introduction situe clairement le contexte scientifique de la thèse en exposant les difficultés rencontrées par les méthodes fondées sur l'arithmétique par intervalles dans le domaine de la robotique mobile embarquée, notamment les limitations sévères des implémentations logicielles dues aux changements fréquents de mode d'arrondi et à l'absence de fonctions élémentaires correctement arrondies sur les processeurs conventionnels. Elle présente ensuite de manière succincte les principales contributions du travail : d'une part, un raffinement théorique de la théorie des contracteurs via l'introduction des contracteurs totaux et du drapeau logique *iota* (ι) permettant de gérer rigoureusement les fonctions partiellement définies ; d'autre part, la conception et l'implémentation d'une extension ISA personnalisée nommée *xinterval*, pour l'architecture RISC-V, offrant une accélération matérielle dédiée des opérations d'arithmétique d'intervalles. Ces apports sont positionnés par rapport à l'état de l'art en soulignant à la fois l'originalité de l'approche matérielle et son potentiel pour surmonter les verrous actuels des bibliothèques logicielles existantes.

Chapitre 1 - Méthodes par intervalles appliquées à la robotique

Ce chapitre introductif pose les fondements théoriques de l'analyse par intervalles et de son application aux problèmes de robotique. Pierre Filiol y développe, avec une pédagogie appréciable, les concepts de base : définition des intervalles, arithmétique par intervalles, vecteurs d'intervalles, contracteurs (notamment HC4R), sous-pavages et algorithme SIVIA. Il illustre ces notions au moyen d'un exemple complet de localisation d'un robot mobile dans un environnement partiellement connu.

Le chapitre se conclut par un tour d'horizon du standard IEEE-1788 et des principales bibliothèques logicielles existantes (GAOL, FILIB, MPFI, etc.), ainsi que par une analyse critique des limitations de performance propres aux implémentations logicielles sur des architectures généralistes. En particulier, la problématique du changement fréquent de mode d'arrondi, imposé par la norme IEEE-754, vide le pipeline d'instructions et constitue le principal goulot d'étranglement motivant la suite de la thèse.

Chapitre 2 - Stratégies d'accélération matérielle pour l'arithmétique d'intervalles

Ce chapitre propose un panorama structuré des stratégies d'accélération matérielle applicables aux algorithmes fondés sur l'arithmétique par intervalles. Pierre Filiol y distingue deux niveaux d'intervention : d'une part, l'accélération au niveau du graphe d'exécution, illustrée par l'exploration des architectures GPU via CUDA, dont les limites sont analysées dans le contexte spécifique des méthodes par intervalles ; d'autre part, l'accélération au niveau des contracteurs élémentaires, pour laquelle sont examinées les approches reposant sur les instructions SIMD ainsi que sur la conception de matériel dédié.

L'auteur montre de manière argumentée que les dépendances de données inhérentes aux algorithmes de type SIVIA, combinées aux contraintes liées à la gestion des modes d'arrondi, restreignent fortement l'intérêt des GPU dans ce cadre. Il en déduit la pertinence d'une approche fondée sur les FPGA, qui offrent un contrôle fin des opérateurs et des politiques d'arrondi, et identifie notamment l'outil FloPoCo comme une solution particulièrement prometteuse pour le développement de systèmes embarqués spécialisés.

Chapitre 3 - Architecture RISC-V

Le troisi me chapitre introduit le standard RISC-V, ses motivations historiques (passage des architectures CISC vers RISC, limites des jeux d'instructions dits « legacy ») et sa philosophie modulaire. Il d crit l'architecture du jeu d'instructions de base (RV32I), le syst me d'extensions standard, les conventions d'appel et les diff rents formats d'instructions. Une attention particuli re est port e aux extensions flottantes F et D ainsi qu'  l'encodage du mode d'arrondi dans les instructions, propri t  essentielle pour l'arithm tique d'intervalles.

Le chapitre aborde  galement la nomenclature des processeurs RISC-V, la notion de soft-core et les possibilit s d'extensions personnalis es, pr parant ainsi le terrain pour la d finition de l'extension xinterval.

Chapitre 4 - G n ration exacte du compl mentaire d'un contracteur

Ce chapitre constitue la premi re contribution th orique majeure, consacr e   la g n ration exacte du compl mentaire de contracteurs lorsque interviennent des fonctions partiellement d finies (par exemple $\sqrt{\cdot}$ ou $\log(\cdot)$). L'introduction d'un drapeau logique sp cifique, l' l ment *iota*, permet de signaler une violation de domaine sans interrompre le flux de calcul, garantissant ainsi la continuit  et la rigueur de l'analyse.

Ces id es sont  tendues aux intervalles, avec la notion d'intervalles totaux, ainsi qu'aux contracteurs, qui donnent lieu   la d finition de contracteurs totaux. Un drapeau d di  permet de tracer les violations de domaine au cours des propagations forward-backward. Des exemples illustrent la construction de l'AST associ    une contrainte et   son compl mentaire, et les pavages obtenus d montrent la correction de l'approche.

Chapitre 5 - Vue haut niveau de l'extension Xinterval

Ce chapitre d crit la conception de l'extension ISA xinterval pour RISC-V. Il pr cise le choix de l'encodage des intervalles dans les registres flottants (NaN boxing, format Float31, repr sentation de l'intervalle vide) ainsi que la s mantique des instructions permettant de cr er, charger, stocker et interroger des intervalles. Un ensemble d'instructions de base est propos  pour les op rations ensemblistes (union, intersection) et pour les half-contractors fond s sur les op rateurs arithm tiques (+, -,  ,  ) et sur diverses fonctions r elles (sqrt, exp, log, sin, cos, asin, acos, etc.).

Le chapitre montre comment ces primitives s'int grent dans le sch ma forward-backward des contracteurs, et un tableau d taille l'usage des op rations ensemblistes pour la propagation arri re (backward).

Chapitre 6 - Micro-architecture de l'extension xinterval

Ce chapitre pr sente la micro-architecture VHDL de xinterval et son int gration dans un c ur RISC-V. Il d crit l'interface mat riel-logiciel (d codage des instructions de type R, interface commune aux op rateurs) puis le fonctionnement d'un « minimal xinterval core » reli  au pipeline RISC-V.

Les diff rents modules mat riels (itv_make, itv_extract, itv_set, itv_add, itv_mul, itv_div, itv_sqrt, itv_exp, itv_log, itv_trig, itv_trig_inv) sont expliqu s   partir de leurs entit s top-level et de leurs datapaths, en montrant comment les blocs FloPoCo g rent les arrondis et les cas particuliers.

Le chapitre donne enfin les r sultats de synth se sur FPGA, notamment sur le XC7Z020   100 MHz, avec un tableau r capitulant les ressources utilis es et les fr quences obtenues.

Chapitre 7 - M thodologie de conception, validation et benchmarking

Le dernier chapitre pr sente la m thodologie utilis e pour d velopper, valider et  valuer les performances de xinterval. Il d crit une plate-forme de test de type *hardware-in-the-loop*, o  un simulateur d'ISA RISC-V interagit avec le c ur xinterval r el via plusieurs wrappers successifs (cosimulation GHDL, communication UART, int gration AXI/Zynq), en expliquant les avantages et les limites de chacune de ces solutions.

Le chapitre d taille ensuite le flot de d veloppement applicatif, de la compilation avec GCC jusqu'  l'ex cution sur le SoC simul , et propose un mod le de la latence du simulateur. Un ensemble de benchmarks robotiques, fond s sur des contracteurs de complexit  croissante, est utilis  pour comparer une plate-forme RISC-V standard   une plate-forme  quip e de xinterval, selon une m thodologie garantissant l' quit  des mesures.

Les r sultats montrent des acc l rations comprises entre x2,7 et x7,8, et plusieurs figures illustrent la pr cision des sous-pavages obtenus.

Bibliographie : La th se s'appuie sur un corpus de 134 r f rences dont la pertinence est av r e, couvrant efficacement l'ensemble des domaines abord s dans le manuscrit de th se.

  Angers, le 2 avril 2026



Mehdi Lhommeau

Professeur des Universit s

LARIS, Polytech Angers

mehdi.lhommeau@univ-angers.fr