

Performances de l'Intelligence Artificielle dans les Systèmes Embarqués Hétérogènes

Laboratoire :	LIAS (https://www.lias-lab.fr)
Établissement :	ISAE-ENSMA (https://www.ensma.fr)
Directeur de thèse :	Emmanuel Grolleau
Co-encadrants académiques :	Henri Bauer, Frédéric Ridouard
Entreprise :	AC6 (https://www.ac6.fr/)
Référénts industriels :	Roy Jamil, Ayoub Bourjilat
Financement :	CIFRE AC6

1 Candidature recherchée

Le ou la candidat(e) doit être titulaire d'un diplôme d'ingénieur ou Master, et avoir de solides compétences en informatique, programmation, architecture. La thèse se déroulera principalement dans les locaux d'AC6 à Courbevoie, avec des séjours au LIAS, ISAE-ENSMA à Chasseneuil du Poitou, site du Futuroscope.

Le dossier de candidature doit être envoyé par courrier électronique **au plus tard le 15/09/2024, minuit**, à grolleau@ensma.fr et est constitué de :

- Lettre de motivation;
- CV mentionnant explicitement le parcours académique;
- feuilles de notes des années universitaires mentionnant, si possible, le classement;
- tout élément jugé utile pour mettre la candidature en valeur.

Nous recherchons en particulier :

- Connaissances de base en microcontrôleurs ou microprocesseurs,
- Connaissance en systèmes d'exploitation RTOS et Linux embarquée,
- Connaissance approfondie en machine learning, réseaux neuronaux (LSTM, transformers, etc.) et deep learning (MobileNet, etc.),
- Maîtrise des outils et langages de programmation tels que C/C++, TensorFlow/KERAS, Matlab et Python,
- Maîtrise du français et de l'anglais, tant à l'écrit qu'à l'oral,
- L'envie d'enseigner et de partager des connaissances.

2 Contexte et problématique

La plupart des drones civils de taille micro-drone ou mini-drone reposent sur des autopilotes open source sur étagère (PX4 [PX4], ArduPilot [Ard], Paparazzi [Pap], etc.). Bien que prévus pour être extensibles au besoin, ces autopilotes, et les fonctions avancées qu'ils peuvent proposer, sont complexes à valider et pour le moment difficiles, voire impossibles, à certifier suivant des normes avioniques de sûreté de fonctionnement. L'une des difficultés est que ceux-ci sont très complexes, puisque constitués de plus de 1500 fichiers C/C++, de plus de 500k lignes de codes, et de milliers de fichiers de configuration [Kam+23]. En effet, les autopilotes sur étagère sont en fait des générateurs d'autopilotes [Hat+23], capables de fonctionner sur des géométries



de plateformes (voilure fixe, voilure tournante, multi-rotor, rover, bateau, sous-marin, plus léger que l'air, etc.), et prenant en compte des centaines de capteurs différents, dont la liste évolue continuellement.

En plus de la complexité du code d'un autopilote, la validation du système d'exploitation pose un autre challenge. Différents autopilotes tels que CogniPilot [Cog] proposent d'utiliser un système d'exploitation temps réel (RTOS) sûr, tel que Zephyr RTOS [Zep], cependant, leur portabilité est pour le moment très limitée. Des autopilotes plus versatiles, comme Paparazzi, sont capables de s'exécuter sans RTOS sous-jacent (*bare-metal*), mais ils profitent d'un écosystème moins développé que les autopilotes plus connus, tels que PX4. Des fonctions de plus en plus avancées sont en effet disponibles avec le développement de l'intelligence artificielle (IA), notamment fonctions de navigation basées vision, reconnaissance de la parole, des personnes, des gestes, etc. L'utilisation de matériel spécifique embarqué permettant l'inférence en temps réel de modèles entraînés pour ces fonctions offre opportunités et challenges, notamment sur le temps de réponse de ces applications d'une part, et de leur colocalisation, ou non, sur les mêmes ressources de calcul que l'autopilote. La complexité de la partie RTOS, de la partie autopilote, et surtout de fonctions avancées notamment basées IA, va croissante et rend la validation des autopilotes plus complexes à mesure qu'ils sont plus évolués.

Une alternative à la validation et certification d'un autopilote complexe est de valider un ensemble constitué d'un autopilote complexe utilisant potentiellement l'IA, et d'un autopilote de secours, qui est minimaliste, s'exécute sur RTOS pouvant être validé, ou sans RTOS. L'idée est que tant qu'aucun défaut majeur n'est constaté sur l'autopilote complexe, celui-ci offre toutes les fonctionnalités qu'il embarque, sous la surveillance de l'autopilote de secours. En cas de défaut empêchant l'autopilote de contrôler le drone, l'autopilote de secours prend la main, offrant moins de fonctionnalités, mais permettant à minima un retour à la base.

3 Problématiques scientifiques

Comme nous l'avons montré dans [Hat+23], les implémentations d'autopilotes possèdent des spécificités : la partie contrôle d'attitude (assiette, angle d'attaque, vitesse), de l'acquisition des capteurs nécessaires à l'estimation d'état au contrôle des actionneurs, est généralement implémentée dans une unique tâche périodique, agissant comme un ordonnanceur statique pour les fonctions qui y sont intégrées. Cette tâche est en concurrence avec diverses tâches chargées d'opérer les entrées-sorties, et les tâches dites de charge utile, qui embarquent des fonctions spécifiques à la réalisation de la mission. Ces autres tâches peuvent être périodiques ou sporadiques, par conséquent, il y a un mélange entre tâches basées sur le temps et tâches sporadiques sur des processeurs de plus en plus sollicités (vision, inférence de modèles d'IA, etc.). La coexistence sur un cœur de calculs, de tâches basées sur le temps et de tâches basées sur les événements a été étudiée initialement dans [PL05]. Cependant, les travaux concernent des tâches indépendantes.

Un nombre croissant de travaux aborde le problème de l'inférence sur réseaux de neurones dans des systèmes embarqués [Li+20; Wan+23]. Cependant, ils considèrent des systèmes de tâches qui ne sont pas représentatifs du comportement ni des architectures embarquées sur les drones. Cette thèse prendra en considération les spécificités des systèmes de tâches embarqués sur drone, des architectures hétérogènes, en utilisant au mieux les accélérateurs matériels.

De plus, une telle étude de cas offre de nombreux points d'entrée pédagogiques pour l'enseignement des systèmes embarqués temps réel, points sur lesquels la thèse s'appuiera pour



illustrer des concepts de conception, développement, validation, architecture, systèmes d'exploitation, etc.

4 Travaux de la thèse

Le ou la doctorant(e) suivra les différentes étapes de la thèse :

- État de l'art sur l'IA dans les systèmes embarqués :
 - Planification et ordonnancement des ressources utilisées pour l'inférence, sur architectures multicœurs hétérogènes et/ou distribuées.
 - Compression de modèles de réseaux neuronaux pour les systèmes embarqués à faibles ressources.
 - Entraînement sur appareils embarqués [AMI23].
 - Déploiement d'applications IA dans les systèmes embarqués avec des contraintes de temps réel et de sécurité, comme sur Zephyr [Lin24].
- Déploiement d'une application test, composée d'un autopilote sur étagère, et de fonctions avancées basées IA, sur carte munie d'accélérateurs matériels.
- Modélisation en vue de la validation sur architecture hétérogène, en prenant compte le caractère mixte des tâches (basées sur le temps ou sur les événements). Les principales contributions académiques sont attendues sur cet aspect.
- Mise en place de supports pédagogiques sur l'IA embarquée, basés sur cette étude de cas.

Références

- [AMI23] Ghattas AKKAD, Ali MANSOUR et Elie INATY. « Embedded deep learning accelerators : A survey on recent advances ». In : *IEEE Transactions on Artificial Intelligence* (2023).
- [Ard] ARDUPILOT. *ArduPilot*. en. URL : <https://ardupilot.org> (visité le 14/05/2024).
- [Cog] COGNIPILOT. *Airy Dev Guide*. URL : <https://airy.cognipilot.org/> (visité le 14/05/2024).
- [Hat+23] Gautier HATTENBERGER, Fabien BONNEVAL, Matheus LADEIRA, Emmanuel GROLLEAU et Yassine OUHAMMOU. « Design of Micro-drone Autopilot Architecture with Static Scheduling Optimization ». en. In : *Unmanned systems* (oct. 2023), p. 1. DOI : [10.1142/S2301385024430015](https://doi.org/10.1142/S2301385024430015). URL : <https://enac.hal.science/hal-04357701>.
- [Kam+23] Soulimane KAMNI, Antoine BERTOUT, Emmanuel GROLLEAU, Gautier HATTENBERGER et Yassine OUHAMMOU. « Easing the tuning of drone autopilots through a model-based framework ». In : *Journal of Computer Languages* 77 (nov. 2023), p. 101240. ISSN : 2590-1184. DOI : [10.1016/j.col.2023.101240](https://doi.org/10.1016/j.col.2023.101240). URL : <https://www.sciencedirect.com/science/article/pii/S2590118423000503>.
- [Li+20] En LI, Liekang ZENG, Zhi ZHOU et Xu CHEN. « Edge AI : On-Demand Accelerating Deep Neural Network Inference via Edge Computing ». en. In : *IEEE Transactions on Wireless Communications* 19.1 (jan. 2020), p. 447-457. ISSN : 1536-1276, 1558-2248. DOI : [10.1109/TWC.2019.2946140](https://doi.org/10.1109/TWC.2019.2946140). URL : <https://ieeexplore.ieee.org/document/8876870/> (visité le 27/06/2024).
- [Lin24] LINARO. *Linaro/zephyr_confidential_ai*. original-date : 2022-09-23T11 :30 :45Z. Juin 2024. URL : https://github.com/Linaro/zephyr_confidential_ai (visité le 27/06/2024).



- [Pap] PAPARAZZI. *PaparazziUAV*. URL : <https://paparazzi-uav.readthedocs.io> (visité le 14/05/2024).
- [PL05] Rodolfo PELLIZZONI et Giuseppe LIPARI. « Feasibility Analysis of Real-Time Periodic Tasks with Offsets ». en. In : *Real-Time Systems* 30.1 (mai 2005), p. 105-128. ISSN : 1573-1383. DOI : [10.1007/s11241-005-0506-x](https://doi.org/10.1007/s11241-005-0506-x). URL : <https://doi.org/10.1007/s11241-005-0506-x> (visité le 15/05/2024).
- [PX4] PX4. *PX4 - Open Source Autopilot for Drones*. en-US. URL : <https://px4.io/> (visité le 14/05/2024).
- [Wan+23] Ruiqi WANG, Hanyang LIU, Jiaming QIU, Moran XU, Roch GUÉRIN et Chenyang LU. « Progressive Neural Compression for Adaptive Image Offloading Under Timing Constraints ». English. In : IEEE Computer Society, déc. 2023, p. 118-130. ISBN : 9798350328578. DOI : [10.1109/RTSS59052.2023.00020](https://doi.org/10.1109/RTSS59052.2023.00020). URL : <https://www.computer.org/csdl/proceedings-article/rtss/2023/285700a118/1Ujijk4tRHG> (visité le 27/06/2024).
- [Zep] ZEPHYR. *The Zephyr Project – A proven RTOS ecosystem, by developers, for developers*. URL : <https://zephyrproject.org/> (visité le 14/05/2024).

