Stage : Définition et développement de modules logiciels dans un framework multi-process

Proposition de stage 2026

Portage de MurMuRe en C++ dans NVIDIA Holoscan et état de l'art pour le nettoyage de spectres dynamiques en temps réel

L'Observatoire Radioastronomique de Nançay (ORN), unité de l'Observatoire de Paris et UAR du CNRS, développe et opère des instruments radio pour l'astronomie et l'astrophysique. Ses missions couvrent la R&D instrumentale, la participation aux grands projets internationaux (p. ex. SKA, LOFAR) et l'exploitation scientifique des instruments du site de Nançay (Grand Radiotélescope Décimétrique, NenuFAR, Radiohéliographe, Réseau décamétrique).

Sujet de stage :
Contexte. MurMuRe (*Modular Multicast Receiver*) est **déjà en production** sur plusieurs radiotélescopes de l'ORN. Toutefois, sa nature *pythonnesque* impose un **plafond de débit temps réel** $<$ **10 Gbit/s**. Pour dépasser cette limite, l'architecture visée s'oriente vers des **technologies direct-to-GPU** (p. ex. DPDK/GPUDirect) et une **réécriture C++**. L'écosystème **NVIDIA Holoscan** simplifie largement cette transition (graphe d'opérateurs, gestion mémoire zéro-copie, intégration IA). *Le présent stage se concentre strictement sur le* ***data_pipeline*** *MurMuRe et son portage C++/Holoscan* (les couches de réception réseau sont déjà développées).

Objectif général. (A) Porter et intégrer en **C++** sous **Holoscan** les *processors* de la chaîne **data_pipeline** MurMuRe (du pré-traitement aux produits temps réel), en reproduisant le graphe et les interfaces de données. **(B)** Produire un **état de l'art** et un **débroussaillage** des approches IA pour le *nettoyage de spectres dynamiques* (RFI/artefacts), sans développement complet.

Découpage du stage

Projet A — Portage C++/Holoscan du *data_pipeline* MurMuRe (focus pipeline, sans receive

- **Cartographier** le *data_pipeline* : liste des processeurs existants, dépendances, types/tailles de buffers, cadence...
- **Définir** les **interfaces** entre les *operators* Holoscan : shapes temporelles/fréquentielles/polarisation, formats (réel/complexe, 8/16/32 bits), métadonnées...
- **Implémenter** des *operators* Holoscan C++ correspondants :
  ▷ pré-traitements/reshape ▷ canalisation (FFT/iFFT) ▷ puissances & produits de Stokes
  ▷ intégrations/accumulations ▷ génération de spectrogrammes
  ▷ *writers* de fichiers "raw", "spectra" et "filterbank".
- **Composer** une application Holoscan reproduisant le graphe *data_pipeline* (CMake, Docker) et **documenter** le câblage (diagramme du flux).
- **Tester/benchmarker** : validation fonctionnelle, **latence bout-en-bout**, stabilité longue durée, profilage GPU/CPU ; **tests/validation à haut débit** $>$ **10 Gbit/s** sur des données réels.
- **Livrables** : modules C++ documentés, application Holoscan minimale, scripts d'exécution/profiling, CI basique (tests unitaires), **rapport technique** du portage.

Projet B — RFI mitigation IA (sans dev complet) : état de l'art + débroussaillage

- **État de l'art ciblé** (10–15 pages) : familles de méthodes (filtres classiques vs DL type U-Net), contraintes temps réel en radio.
- **Jeu de données & protocole** : sources internes/publiques, plan d'annotation minimal, métriques (SNR, préservation des signaux astrophysiques...), protocole d'évaluation.
- **Livrables** : revue bibliographique annotée, tableau comparatif des méthodes, **mémo de RFI mitigation IA**.

Plan indicatif (4–6 mois)

- S1–S2 : prise en main MurMuRe (*data_pipeline*) / Holoscan, audit des processors.
- S2–S7 : portage des processeurs prioritaires, application Holoscan fonctionnelle, premiers benchmarks.
- S7–S11 dev des tests unitaires et validation avec différentes sources de données.
- (léger, en parallèle) : **état de l'art RFI mitigation IA**, **note de cadrage & roadmap** (sans dev complet).
- S11–S16 : stabilisation, documentation, démonstration et **rapport final** (incluant la synthèse IA).

Profil du candidat

- Élève-ingénieur/Master en informatique/traitement du signal/HPC ; C++ et Python requis.
- Notions GPU (CUDA/CuPy), Linux, CMake, Docker, Git.
- Intérêt pour l'instrumentation radio et le *stream processing* temps réel.

Conditions du stage

Le stage se déroulera à l'Observatoire Radioastronomique de Nançay (Sologne). Encadrement mixte ingénieurs/chercheurs ORN/LESIA. Hébergement possible sur site ; restaurant d'entreprise au tarif étudiant.

Personnes à contacter

Louis Bondonneau `<louis.bondonneau@obs-nancay.fr>`
Secrétariat ORN `<secretariat@obs-nancay.fr>` Standard : +33 (0)2 48 51 82 41

Références et ressources

[1] MurMuRe — **data_pipeline** : https://gitlab.obspm.fr/receiver/modular_multicast_receiver/-/tree/main/data_pipeline

[2] HoloHub/Holoscan — dépôt : https://gitlab.obspm.fr/receiver/holohub

[3] NenuFAR — site officiel : https://nenufar.obs-nancay.fr/