5 Description de la démarche suivie et des travaux réalisés

5.1 Démarche expérimentale

5.1.1 Validation par simulation virtuelle

La démarche pour valider les observateurs par simulation est illustrée à la Figure 3 et suit une approche modulaire centrée sur la modélisation du peloton, le contrôle coopératif, l'observation distribuée et la validation sur plateforme Quanser QCar2/QLabs. Cette approche vise à garantir la robustesse, la fiabilité et la reproductibilité des algorithmes dans des conditions simulées proches du réel :

- **Phase 1 – Mise en place de l'environnement** : configuration du cadre virtuel sous Ubuntu et déploiement du conteneur Quanser Virtual Environment pour connecter QLabs et QCar2.
- **Phase 2 – Communication V2V et architecture logicielle** : mise en place d'une communication inter-véhicules via UDP et d'une architecture Python modulaire.
- **Phase 3 – Contrôle longitudinal et latéral :** deux boucles de commande ont été implémentées :
  - Longitudinale, pour la régulation de la vitesse et de la distance inter-véhicules ;
  - Latérale, pour le maintien de la trajectoire ;
  - Ces contrôleurs reposent sur le modèle coopératif (CACC) et s'adaptent aux perturbations de communication.
- **Phase 4 – Développement d'un observateur et d'un système de confiance** : un observateur distribué a été développé pour estimer les états des véhicules du peloton, même en présence de retards ou de pertes d'informations. Un système de confiance (trust system) pondère la fiabilité des données reçues et ajuste dynamiquement la fusion d'informations.
- **Phase 5 – Validation sur QLabs/QCar2** : L'intégration complète du système a été réalisée et testée en simulation temps réel sous QLabs. La phase de transfert vers les **scénarios physiques sur QCar2 réel est en cours de préparation**, afin de valider expérimentalement la robustesse des algorithmes dans un environnement réel.

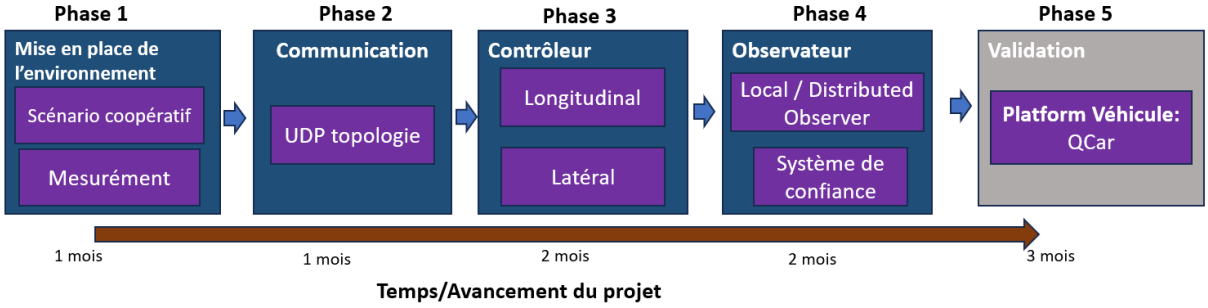

Figure 3 : Démarche expérimentale simulation

5.1.2 Préparation aux essais en environnement réel

L'objectif principal est de concevoir un capteur embarqué intelligent capable d'estimer des grandeurs non mesurées et de détecter les perturbations liées aux cyberattaques ou aux défauts capteurs. Ce capteur, renforcé par des algorithmes d'estimation avancés, agit comme

un capteur logiciel : il associe les mesures radar aux véhicules, reconstruit les variables critiques et distingue les anomalies physiques des attaques malveillantes.

- **Phase 1 – Définition de l'architecture :**
  - Cette première phase a porté sur l'analyse des besoins fonctionnels et la définition de l'architecture technique de la carte embarquée.
  - Elle a permis d'identifier les capteurs nécessaires, les microcontrôleurs à utiliser, ainsi que les protocoles de communication adaptés (SPI, I²C, UART, Wi-Fi 6).
  - Les travaux de cette phase ont abouti à la conception du schéma bloc de la carte et à la sélection des composants clés, préparant ainsi la conception électronique détaillée.
- **Phase 2 – CAO pour l'électronique :**
  - Cette étape a consisté à traduire l'architecture en un design électronique complet sous KiCad.
  - Elle a inclus la réalisation des schémas électriques, le routage et le placement des composants sur un PCB 4 couches.
  - Un travail approfondi a été mené sur l'alimentation, les contraintes d'impédance, la compatibilité électromagnétique et la gestion des signaux haute fréquence (Wi-Fi, GNSS).
- **Phase 3 – Prototypage et programmation embarquée :**
  - Dans cette phase, les premières validations logicielles ont été réalisées à l'aide de kits de développement avant l'intégration sur carte réelle.
  - Le développement du firmware, sous STM32CubeMX et STM32CubeIDE, a permis de configurer les protocoles de communication, d'intégrer un système d'exploitation temps réel (RTOS) et de préparer le code d'acquisition et de communication inter-véhicules.
- **Phase 4 – Tests et protocoles de validation :**
  - La dernière phase a concerné la vérification expérimentale du prototype à travers des tests d'alimentation, de communication et de transfert de données.
  - Des protocoles rigoureux ont permis de contrôler la stabilité électrique, la continuité des signaux et la fiabilité des communications UART, I²C, SPI et Wi-Fi.

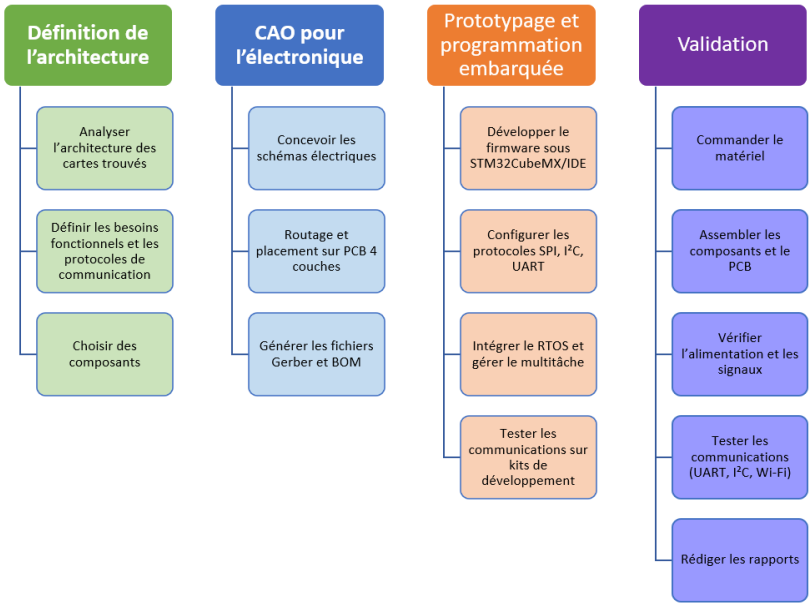

Figure 4 : démarche expérimentale de développement de la carte embarquée

5.2 Travaux & résultats de l'opération de R&D

Les travaux présentés dans cette section s'inscrivent directement dans la continuité de la thèse CIFRE menée entre SEGULA Technologies et le CRAN, consacrée au développement d'algorithmes d'estimation avancés et de solutions embarquées pour renforcer la résilience des véhicules autonomes et connectés. L'année écoulée a été marquée par la mise en œuvre concrète des contributions scientifiques définies dans le sujet de thèse : modélisation du peloton, conception d'observateurs robustes aux entrées inconnues, intégration de mécanismes de confiance et préparation d'une plateforme embarquée dédiée.

Les résultats détaillés ci-dessous traduisent ainsi la progression du projet, depuis les fondements méthodologiques jusqu'aux validations en simulation réaliste et aux premières étapes de l'implémentation matérielle. Ils constituent une étape essentielle avant la transition vers les essais expérimentaux prévus sur QCar2 et sur systèmes réels.

~~5.3.1~~5.2.1 Simulations virtuelles

~~5.3.1.1~~5.2.1.1 Mise en place de l'environnement

Cette étape est fondamentale, car l'ensemble des expériences et validations ultérieures doivent être réalisées dans ce cadre virtuel. Elle constitue ainsi la pierre angulaire du projet et garantit la cohérence ainsi que la reproductibilité des essais. La configuration générale est illustrée à la Figure 5.

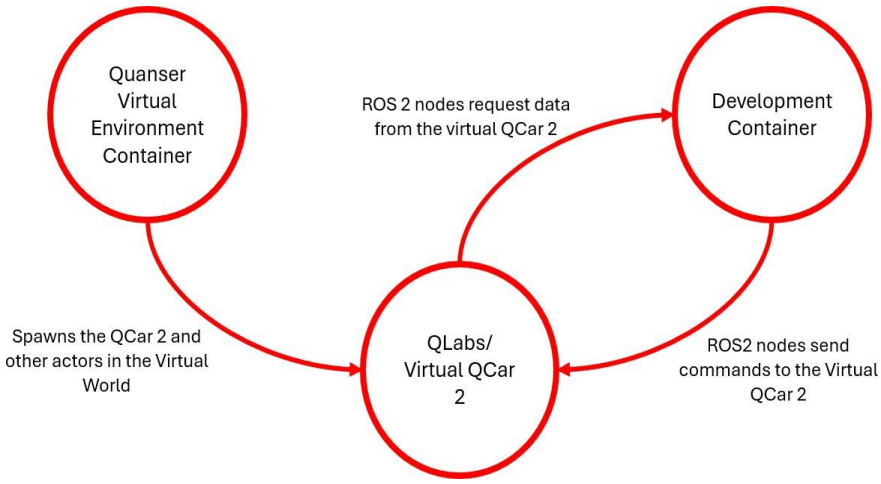

Figure 5 : Configuration de l'environnement virtuel QLabs avec QCar2

Dans cet environnement virtuel, deux véhicules ont été configurés : un véhicule leader chargé de suivre une trajectoire prédéfinie et un véhicule follower destiné à reproduire le comportement coopératif du peloton. La trajectoire du leader a été définie à partir des points de passage (waypoints). Ces points de passage, visibles à la Figure 6, définissent une boucle fermée que le véhicule leader doit parcourir de manière répétée. En fixant une vitesse désirée de 0.3 m/s, nous assurons un mouvement fluide et constant qui sert de référence au véhicule suiveur.

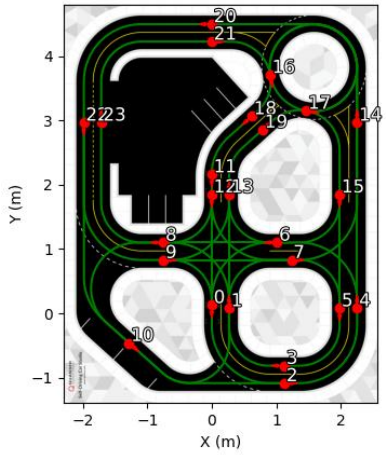

Figure 6 : Points de passage sélectionnés dans la carte de QLabs.

Le scénario inclut le sol, des murs et un passage piéton, ce qui le rapproche des conditions réelles de circulation.

~~5.3.1.2~~5.2.1.2 Mesures disponibles sur QCar2

Pour concevoir le système de confiance, il est essentiel de définir au préalable les mesures et métadonnées disponibles sur le véhicule autonome connecté QCar2. Ces données, qui constituent la base de la détection d'anomalies et de l'évaluation de la fiabilité des voisins, proviennent de trois sources principales : les capteurs embarqués, les modules de perception (fusion LiDAR-caméra) et les échanges de communication V2V. Contrairement à de nombreux travaux s'appuyant sur des jeux de données publics comme KITTI ou A2D2, ce projet utilise des données directement issues de la plateforme matérielle QCar2 et de son environnement de simulation QLabs, garantissant ainsi une validation en conditions réalistes. Le tableau suivant récapitule l'ensemble des mesures exploitées dans le cadre de ce projet.

Catégorie	Mesure / Métadonnée	Source/Méthode d'obtention	Description / Utilisation
Capteurs Embarqués (Bruts)	Position (x,y)& Horodatage	GPS virtuel QLabs (API GPSSync)	Synchronisation temporelle et recalage des trajectoires.
	Accélération & Vitesse angulaire	IMU (accéléromètres, gyroscopes)	Dynamique instantanée du véhicule.
	Nuage de points 2D	LiDAR (RPLiDAR A2, portée 12m)	Détection des obstacles sur 360°.
	Images RGB / Depth	Caméras (Intel RealSense D435 & CSI 360°)	Perception visuelle de l'environnement.
	Vitesse des roues & Angle de braquage	Codeur moteur & Servo de direction	Entrées pour le contrôleur longitudinal et latéral.
Variables Cinématiques (Dérivées)	Position, Orientation (ψ)	QLabs (état global)	État absolu du véhicule dans la carte.
	Vitesse longitudinale (v)	Codeur moteur + Filtre de Kalman	Estimation lissée de la vitesse.
	Accélération (a)	Dérivation de v (Kalman) ou IMU direct	Mesure de l'accélération longitudinale.
	Distance inter-véhicules (d)	Fusion LiDAR-Caméra (pipeline décrit ci-dessous)	Mesure de distance fiable pour le contrôle coopératif.

	Vitesse relative (Δv)	Dérivation temporelle de d	Taux de variation de l'écart avec un véhicule voisin.
Métadonnées V2V	État des voisins (p, v, a)	Messages V2V périodiques	Partage d'information pour la coopération.
	Qualité de la liaison	Taux de perte de paquets, délais	Indicateur de fiabilité de la communication.

Tableau 1 : Synthèse des mesures et métadonnées disponibles sur le QCar2.

Les grandeurs critiques que sont la distance inter-véhicules d et la vitesse relative Δv sont obtenues via un pipeline sophistiqué de fusion LiDAR-caméra, inspiré de (Tim, s. d.) et résumé ci-dessous :

- Segmentation d'image : Utilisation de YOLOv8-seg pour extraire les masques des objets (véhicules, piétons).
- Post-traitement : Application d'une érosion sur les masques pour réduire les faux positifs.
- Projection LiDAR : Projection des points du nuage LiDAR dans le plan de l'image grâce aux matrices de calibration (intrinsèques et extrinsèques).
- Association des données : Les points LiDAR sont associés aux objets segmentés, créant un couplage 2D-3D.
- Regroupement spatial : Un algorithme de clustering (DBSCAN) filtre et regroupe les points fusionnés.
- Estimation 3D : Une Analyse en Composantes Principales (PCA) permet de calculer les boîtes englobantes 3D et d'estimer les positions relatives.
- Calcul des grandeurs finales : La distance d est déduite de la position relative, et vitesse relative est obtenue par dérivation temporelle de d.

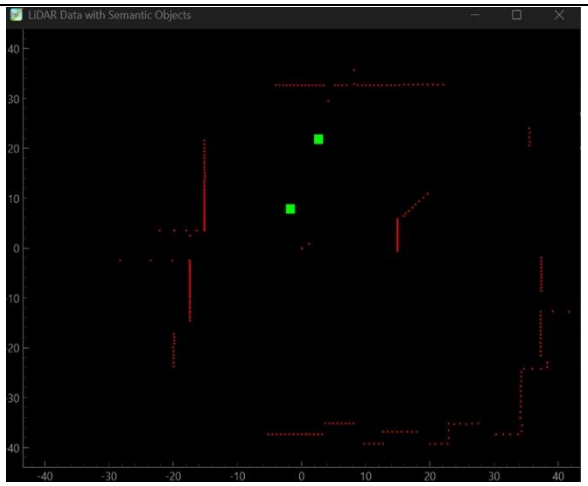

Figure 7 : Vue LiDAR en plongée (bird-eye view) : en rouge les points du nuage LiDAR, en vert les objets détectés et associés aux classes sémantiques.

Figure 8 : Vue caméra avec détection et fusion : boîtes englobantes 2D/3D associées aux véhicules voisins, distances estimées affichées en temps réel.

Ces résultats expérimentaux illustrent le fonctionnement de la chaîne de fusion décrite ci-dessus. La Figure 7 montre comment les points du nuage LiDAR sont regroupés et associés aux objets détectés visuellement, tandis que la Figure 8 illustre la superposition des boîtes englobantes sur la vue caméra avec estimation des distances inter-véhicules (d_2 = 8 m et d_2

= 22 m dans cet exemple). Ainsi, les grandeurs d et Δv utilisées par le contrôleur coopératif et le système de confiance ne proviennent pas seulement de mesures brutes, mais d'une perception multimodale combinant vision et LiDAR.

Ces fondations solides sont désormais en place pour la suite du projet, qui se concentrera sur l'implémentation et la validation des algorithmes avancés de confiance et de coopération.

~~5.3.1.3~~5.2.1.3 Communication V2V et modularisation logicielle

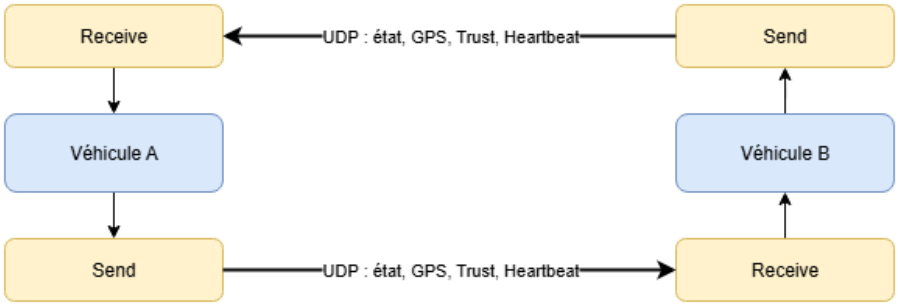

Figure 9 **: V2V communication**

La mise en place du système repose non seulement sur les algorithmes de contrôle et d'estimation, mais aussi sur une architecture logicielle claire permettant la communication robuste entre véhicules virtuels dans QLabs. Chaque véhicule QCar2 est associé à un module « *CommHandler* », responsable de l'échange d'informations avec ses voisins via UDP. À chaque période (100 ms), le véhicule envoie un paquet contenant :

- L'identifiant du véhicule (id) ;
- Sa position (x, y) et son orientation ψ ;
- Sa vitesse v et son accélération a ;
- Un horodatage synchronisé par le serveur GPS virtuel.

Communication V2V : Chaque message est accusé de réception (ACK) afin d'éviter les pertes silencieuses. Un mécanisme de *heartbeat* permet en outre de détecter la disparition d'un voisin (ex. déconnexion ou panne simulée). Ces informations sont directement utilisées par les observateurs distribués et par le système de confiance.

Modularisation logicielle : Afin de garantir la lisibilité et la réutilisabilité du code, le projet a été découpé en plusieurs modules Python interconnectés :

- *md_environment* : gestion de l'environnement QLabs (spawn de la route, obstacles, véhicules).
- *md_vehicle* : encapsulation du comportement d'un véhicule (leader/suiveur), intégrant contrôleur, observateur et communication.
- *md_comm_handler* : gestion bas niveau de la communication V2V (sockets, ACKs, *heartbeats*).
- *md_gps_sync* : synchronisation temporelle par un serveur GPS virtuel.
- *md_logging_config* : gestion centralisée des journaux (logger) pour tracer les états et déboguer.
- *md_main* : script principal orchestrant la simulation (initialisation, lancement, arrêt).

Cette modularisation a facilité les tests incrémentaux : par exemple, il était possible de valider la communication seule avant d'activer les contrôleurs coopératifs.

Intégration avec QLabs : Grâce à cette architecture, chaque QCar2 dans QLabs fonctionne comme un agent autonome capable de communiquer, estimer et contrôler.

L'utilisation de threads distincts pour la communication (envoi/réception) et pour le contrôle assure une exécution parallèle réaliste, proche du fonctionnement d'un système embarqué réel.

En somme, la communication V2V et la modularisation logicielle constituent la colonne vertébrale du projet : elles assurent la cohérence temporelle des informations, la robustesse face aux pertes de messages, et la possibilité de faire évoluer facilement le code vers de nouveaux scénarios (platooning multi-véhicules, attaques simulées, ajout de capteurs).

~~5.3.1.4~~5.2.1.4 Contrôleurs Longitudinal et Latéral

Pour assurer une conduite autonome fluide et sécurisée dans l'environnement QLabs, le véhicule doit être piloté de manière précise sur ses deux axes : longitudinal (vitesse, distance) et latéral (direction, trajectoire). Ces deux contrôleurs agissent de manière complémentaire pour garantir un suivi de trajectoire fiable.

Le contrôle longitudinal repose sur l'algorithme CACC (Cooperative Adaptive Cruise Control), permettant de maintenir un espacement sûr et dynamique entre véhicules en combinant les mesures locales et les informations transmises par les voisins.

$$u_{\mathrm{CACC},i}(k) = \sum_{j=1}^{i-1}[\kappa_s \left(\hat{s}_i^{(j)}(k) - \hat{s}_0^{(i)}(k) - d_{i,j}(k)\right) + \kappa_v \left(\hat{v}_i^{(j)}(k) - \hat{v}_0^{(i)}(k)\right) + \kappa_a \left(\hat{a}_i^{(j)}(k) - \hat{a}_0^{(i)}(k)\right)]$$

Contrôle latéral : Le problème du corner-cutting, c'est-à-dire la tendance des véhicules suiveurs à couper les virages a été traitée à l'aide du contrôleur à regard prolongé **(**Extended Look-Ahead Controller, ELC**)** proposé par (Bayuwindra et al., 2020). L'idée est de définir un point de poursuite virtuel, noté $P*$, situé d'une distance d_{LA} en avant du véhicule précédent et décalé latéralement de d_{lat}. Le véhicule suiveur calcule alors son angle de braquage en fonction de ce point cible plutôt que du simple centre de gravité du véhicule précédent.

Cette approche présente deux avantages majeurs :

- Elle compense les erreurs latérales liées à la géométrie du virage ;
- Elle maintient la distance inter-véhicules effective telle que prescrite par la politique d'espacement (par ex. constant time-gap), et ce indépendamment du fait que le convoi roule en ligne droite ou en virage.

Une comparaison expérimentale met en évidence l'impact du contrôleur « *look-ahead* » sur le phénomène de « *corner-cutting* ». Sans « *look-ahead* », le véhicule suiveur a tendance à couper le virage et à réduire son rayon de trajectoire, ce qui conduit à un décalage par rapport au leader (Figure 10). En revanche, en utilisant un point de poursuite virtuel avancé, le suiveur suit une trajectoire plus fidèle au leader et conserve une distance latérale correcte (Figure 11).

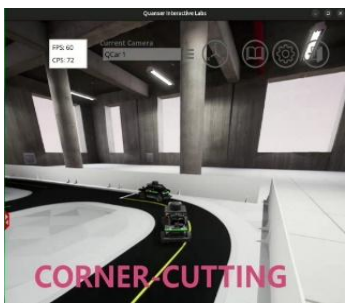

Figure 10 : Exemple de corner-cutting sans contrôleur à regard prolongé.

Figure 11 : Correction du corner-cutting grâce au contrôleur à regard prolongé.

~~5.3.2~~5.2.2 Développement des observateur et système de confiance

L'observateur joue un rôle central dans la reconstitution des états non directement mesurés et dans la robustesse face aux données bruitées ou manquantes. Dans ce projet, deux niveaux d'observation ont été mis en œuvre : un observateur local basé sur les mesures disponibles sur QCar2, et un observateur distribué exploitant la coopération entre véhicules à travers le canal V2V. La figure suivante illustre l'architecture à deux couches : les observateurs locaux fonctionnent au niveau physique, tandis que l'estimation distribuée s'effectue dans une couche virtuelle via les échanges V2V.

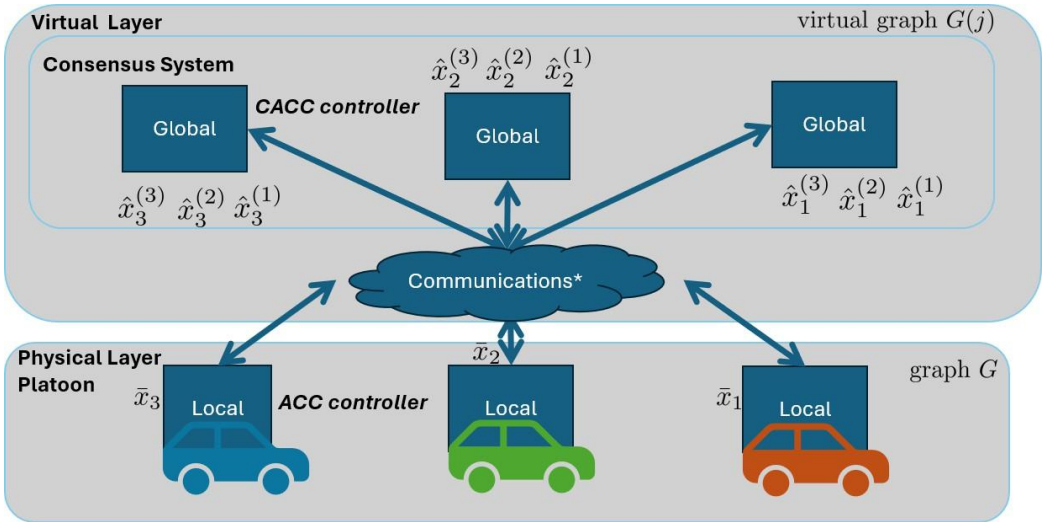

Figure 12 : System Communication protocole

~~5.3.2.1~~5.2.2.1 Observateur local

Chaque véhicule est équipé d'un observateur local (LO_i) capable d'estimer son propre état $x^i(t)$ à partir de ses mesures $y_i(t)$:

$$\hat{x}_0^{(i)}(t+1) = A_b\hat{x}_0^{(i)}(t) + B_b u_i(t) + F_i\left(y_i(t) - C_b\hat{x}_0^{(i)}(t)\right),$$

où F_i est le gain correcteur. Ainsi, $\hat{x}_0^{(i)}(t)$ converge vers l'état réel :

$$\lim_{t\to\infty}\left(\hat{x}_0^{(i)}(t) - x^{(i)}(t)\right) = 0.$$

~~5.3.2.2~~5.2.2.2 Observateur distribué

Pour estimer les états des autres véhicules, on introduit l'observateur distribué (DO). Chaque véhicule i construit une estimation de l'état du véhicule j en s'appuyant sur son observateur local et les informations de ses voisins N_i:

$$\hat{x}_i^{(j)}(k+1) = A\left(\hat{x}_i^{(j)}(k) + \sum_{l\in\ N_i} w_{il}^{(j)}\left(\hat{x}_l^{(j)}(k) - \hat{x}_i^{(j)}(k)\right) + w_{i0}^{(j)}\left(\bar{x}_j(k) - \hat{x}_i^{(j)}(k)\right)\right) + B\hat{u}_j(k), i, j \in V$$

Sous des conditions de connectivité du graphe et de choix approprié des gains w_{il}^j, l'estimation distribuée converge vers l'état réel du système (Nguyen et al., 2025) :

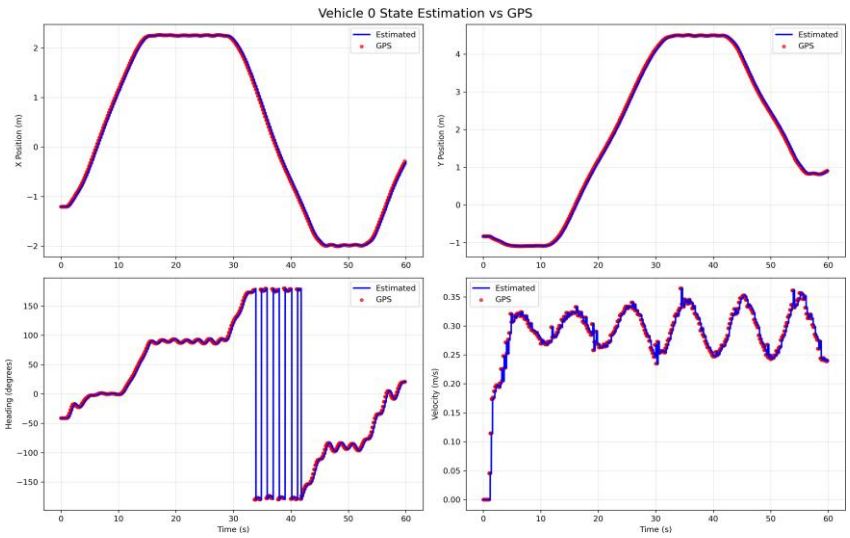

Figure 13 : Comparaison entre l'état estimé et le GPS pour le véhicule 0.

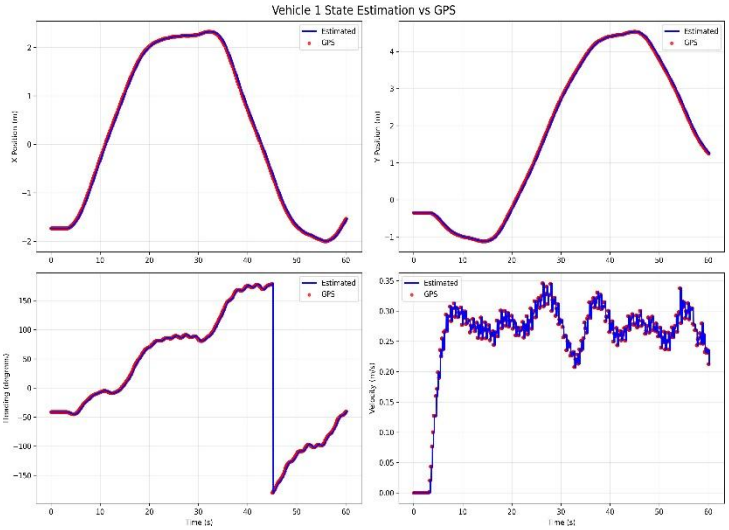

Figure 14 : Comparaison entre l'état estimé et le GPS pour le véhicule 1.

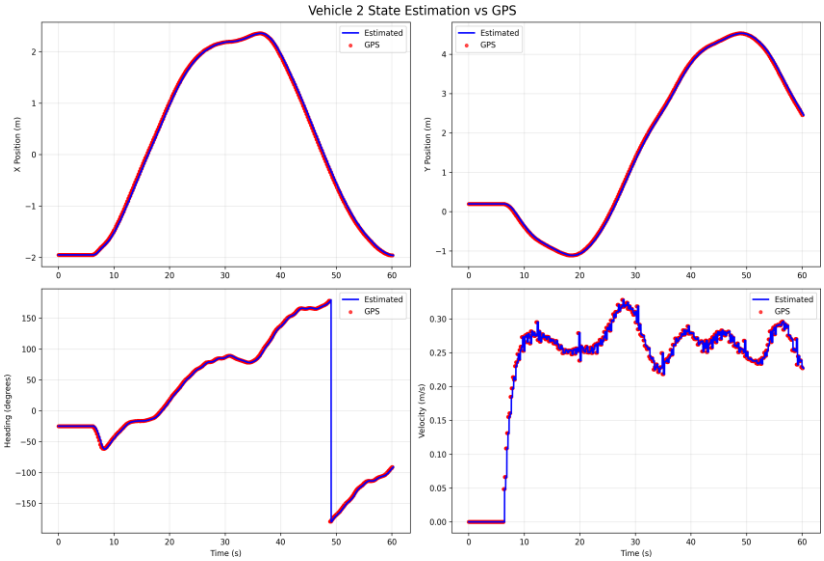

Figure 15 : Comparaison entre l'état estimé et le GPS pour le véhicule 2.

Les Figure **13** à Figure 15 montrent la bonne concordance entre les estimations locales (courbes bleues) et les mesures GPS (points rouges). On observe que les trajectoires (x, y) suivent fidèlement la vérité terrain, et que la vitesse longitudinale est correctement reconstruite malgré le bruit du codeur moteur.

Chaque sous-figure représente une variable d'état différente :

- En haut à gauche : la position x en mètre (axe vertical) en fonction du temps t en seconde (axe horizontal) ;
- En haut à droite : la position y en mètre (axe vertical) en fonction du temps t ;
- En bas à gauche : l'angle de cap (*heading*) en degré (axe vertical) en fonction du temps t ;
- En bas à droite : la vitesse longitudinale v en m/s (axe vertical) en fonction du temps t.

Ces courbes permettent de vérifier simultanément la précision spatiale et cinématique de l'observateur. Dans l'ensemble, ces résultats confirment que l'observateur local est suffisamment précis pour alimenter le contrôleur longitudinal et servir de base à l'observateur distribué.

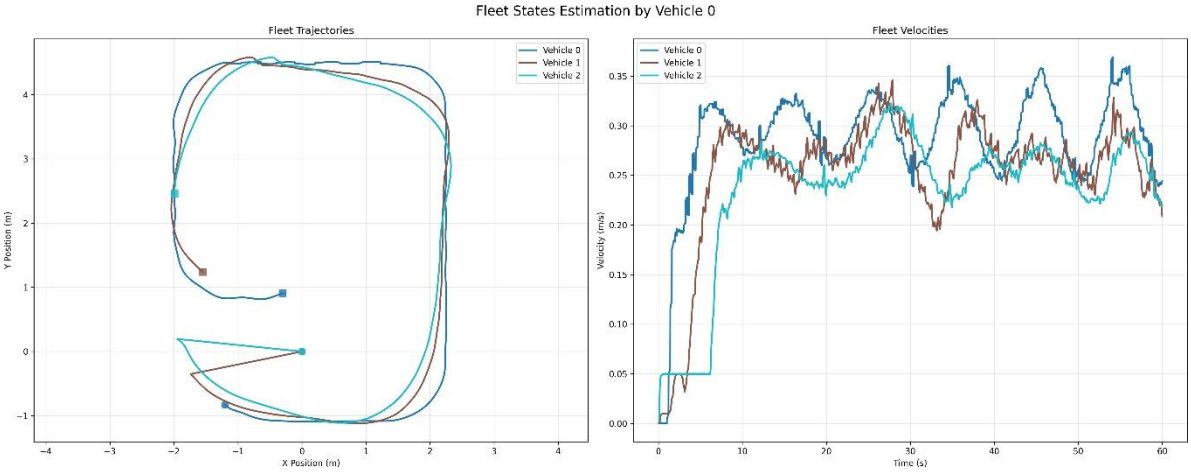

Figure 16 : Estimation distribuée des états de la flotte par le véhicule 0.

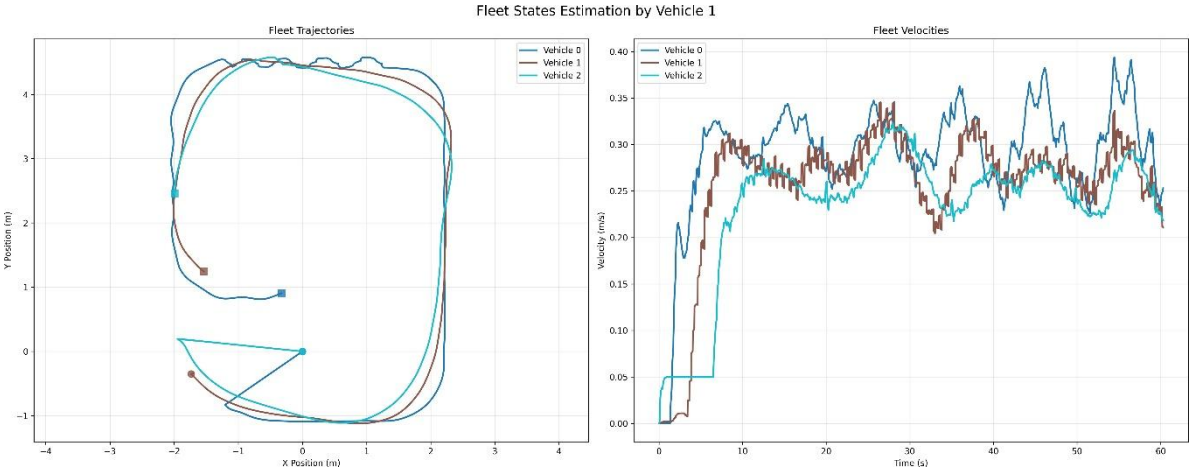

Figure 17 : Estimation distribuée des états de la flotte par le véhicule 1.

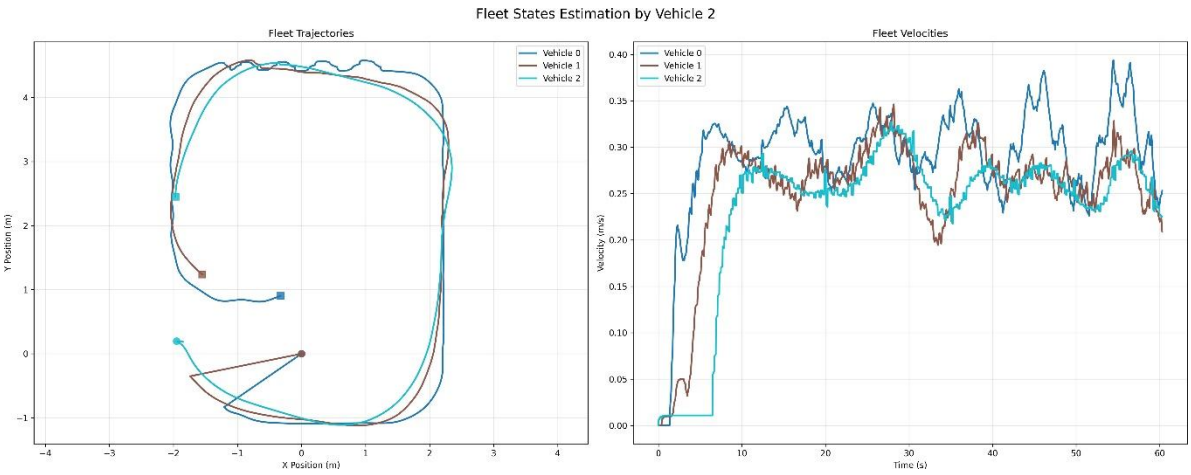

Figure 18 : Estimation distribuée des états de la flotte par le véhicule 2.

Les figures précédentes illustrent la capacité de chaque véhicule à reconstruire l'état global du convoi à partir des échanges V2V. Chaque figure comporte deux sous-parties :

- À gauche : les trajectoires (x, y) de la flotte dans le plan (position *x* en abscisse, position *y* en ordonnée, toutes deux en mètre) ;
- À droite : les vitesses longitudinales v (en m/s, axe vertical) de chaque véhicule en fonction du temps *t* (en secondes, axe horizontal).

On constate que les trajectoires estimées par chaque observateur convergent bien vers des courbes proches, même si elles présentent de légers décalages dus aux délais de communication et à l'imprécision des capteurs locaux. De plus, les vitesses reconstruites par chaque observateur reproduisent correctement la dynamique relative des véhicules.

Il est important de souligner que ces résultats correspondent au cas nominal sans attaque, et servent donc de référence pour les scénarios futurs avec fautes ou perturbations. Chaque observateur distribué est capable d'estimer l'ensemble des véhicules du convoi, et pas seulement son propre état.

Ces résultats mettent en évidence que, même sans mécanisme complet de confiance, l'observateur distribué fournit une estimation robuste et cohérente de la flotte entière, et constitue une base solide pour l'intégration du système de confiance.

En pratique, l'observateur local est alimenté par les mesures du QCar2 (position simulée, vitesse par encodeur, IMU). L'observateur distribué combine ces informations avec les messages V2V reçus. Ainsi, même en cas de bruit de capteur ou de communication défaillante, chaque véhicule conserve une estimation robuste de l'état du convoi.

~~5.3.2.3~~5.2.2.3 Intégration d'un système de confiance (Trust framework)

À partir des mesures locales et des données échangées via la communication V2V, un cadre de confiance est intégré dans l'observateur distribué. L'objectif est d'évaluer la fiabilité de chaque véhicule voisin en fonction de sa cohérence cinématique et de la qualité de ses estimations distribuées :

- **Fonctions d'évaluation :**
  - *evaluate_velocity()* : Vérifie la cohérence entre la vitesse annoncée et le déplacement observé.
  - *evaluate_distance()* : Compare la position rapportée avec la mesure de distance issue du capteur local.

- *evaluate_acceleration()* : Évalue si les variations de vitesse sont dynamiquement plausibles.
- *gamma_cross()* : Mesure la cohérence entre les estimations globales fournies par un véhicule et les données réelles observées.
- *gamma_local()* : Vérifie si les estimations locales des voisins correspondent à la perception du véhicule propre.

- **Confiance locale :**
  - La confiance locale est basée sur les messages directs échangés entre véhicules. Elle combine la vérification de la vitesse, de la distance et de l'accélération pour évaluer si un véhicule est honnête à propos de son propre état *:* LT = $(vitesse \times distance \times accélération)$
- **Confiance globale :**
  - La confiance globale est basée sur les estimations distribuées de la flotte. Elle mesure la fiabilité des informations qu'un véhicule partage sur l'ensemble des autres véhicules : DT = $\gamma_{cross} \times \gamma_{local}$
- **Confiance combinée finale :**
  - La confiance finale combine les aspects local et global afin d'obtenir une mesure complète de la fiabilité d'un véhicule. Un véhicule doit être honnête localement et cohérent globalement pour obtenir un score élevé: $O_j = DT \times LT$
- **Mise à jour du voisinage de confiance**
  - Après calcul du score de confiance, on définit un nouvel ensemble de voisins fiables : $N_{i(t)} = \{ j \in N_i : O_j > 0.5 \}$. Seuls les véhicules dont la confiance dépasse 0.5 sont retenus pour le consensus ou la fusion de données.
- **Poids associés aux voisins fiables**
  - Les poids de communication sont ajustés dynamiquement selon le nombre de voisins fiables :

$$w_{il}^{(j)}(t) = \begin{cases} \frac{1}{n_{w_i}^{G(j)}(t)}, & \text{if } l \in \mathcal{N}_i(t) \cup \{0\} \\ 1 - \sum_{m \in \mathcal{N}_i} w_{im}(t), & \text{if } l = i \\ 0, & otherwise \end{cases}$$

avec $n_{wi}^{G(j)}(t) = \max\{\kappa, |N_i^{G(j)}(t)| + 1\} \geq 1$, pour tout $i \in L$.

Cette formulation garantit un poids normalisé même lorsque peu de voisins sont fiables, et une limite inférieure κ pour éviter la perte totale de connectivité.

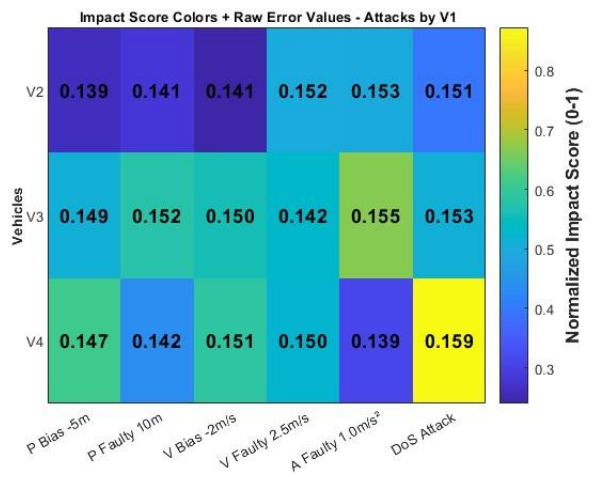

Figure 19 : Carte thermique des scores d'impact normalisés pour l'ensemble des véhicules et des scénarios d'attaque.

Figure 20 : Évolution du score de confiance durant l'attaque DoS (cas 6)

Le tableau ci-dessous récapitule les différents scénarios d'attaque, où le véhicule leader (ID 1) agit comme attaquant entre t = 5 s et 10 s, en ciblant les autres véhicules de la flotte avec des perturbations de type *défaut*, *bias* ou *perte* sur les variables de vitesse et de position.

N° cas	Méthode attaque	Paramètre(s)	Cible (data_type)
1	Biais	-5 m	Position
2	Défaut aléatoire	intensité = 10, p = 0.3	Position
3	Biais	-2 m/s	Vitesse
4	Défaut aléatoire	intensité = 2.5, p = 0.3	Vitesse
5	Défaut aléatoire	intensité = 1.0, p = 0.3	Accélération
6	Perte de données (Drop DoS)	p_drop = 0.5	Position Vitesse Accélération

Tableau 2 : Tableau des scénarios d'attaque

La Figure 19 présente une carte thermique des scores d'impact normalisés pour chaque véhicule et pour l'ensemble des six scénarios d'attaque. Le score d'impact normalisé est obtenu en divisant l'erreur combinée brute par l'erreur maximale observée tous scénarios confondus, ce qui permet de comparer clairement la sévérité relative des attaques.

Deux indicateurs ont été analysés pour quantifier les effets des cyberattaques :
– **l'erreur combinée brute**, reflétant la déviation directe sur la distance, la vitesse et l'accélération estimées ;
– **le score d'impact normalisé**, qui indique la dégradation du système par rapport aux performances nominales.

Parmi l'ensemble des scénarios, l'attaque DoS (Cas 6) présente le score d'impact normalisé le plus élevé (0,872), confirmant qu'il s'agit de l'attaque la plus sévère,

entraînant une forte dégradation de l'estimation et du contrôle sur la période d'attaque. À l'inverse, l'attaque par biais sur la vitesse (Cas 3, biais = -2 m/s) entraîne le score d'impact le plus faible (0,240), ce qui indique un effet limité sur la stabilité générale du peloton.

Concernant la sensibilité des véhicules, le véhicule V4 apparaît comme le plus affecté par les attaques, atteignant le score d'impact maximal (0,872). À l'opposé, le véhicule V2 est le plus résilient, avec le score moyen le plus faible (0,362). Ces observations suggèrent que la dynamique locale ou les dépendances de communication de V4 le rendent plus vulnérable aux perturbations, tandis que V2 conserve des performances plus stables.

De manière générale, les scores d'impact moyens relativement faibles confirment l'efficacité de l'observateur distribué enrichi par le mécanisme de confiance : celui-ci atténue efficacement différents types d'attaques et maintient une estimation précise des états du peloton.

La Figure 20 illustre l'évolution du score de confiance pour l'attaque DoS (Cas 6). Durant l'intervalle 10-15 s, les véhicules V2, V3 et V4 présentent une chute notable du score de confiance sous le seuil de 0.5, révélant la détection correcte de la perturbation. Une fois l'attaque terminée, les scores remontent progressivement, confirmant la capacité du système de confiance à identifier les données malveillantes, à les isoler, puis à restaurer une coopération normale.

~~5.3.3~~5.2.3 Préparation aux essais en environnement réel

La définition de l'architecture de la carte est visible ci-dessous :

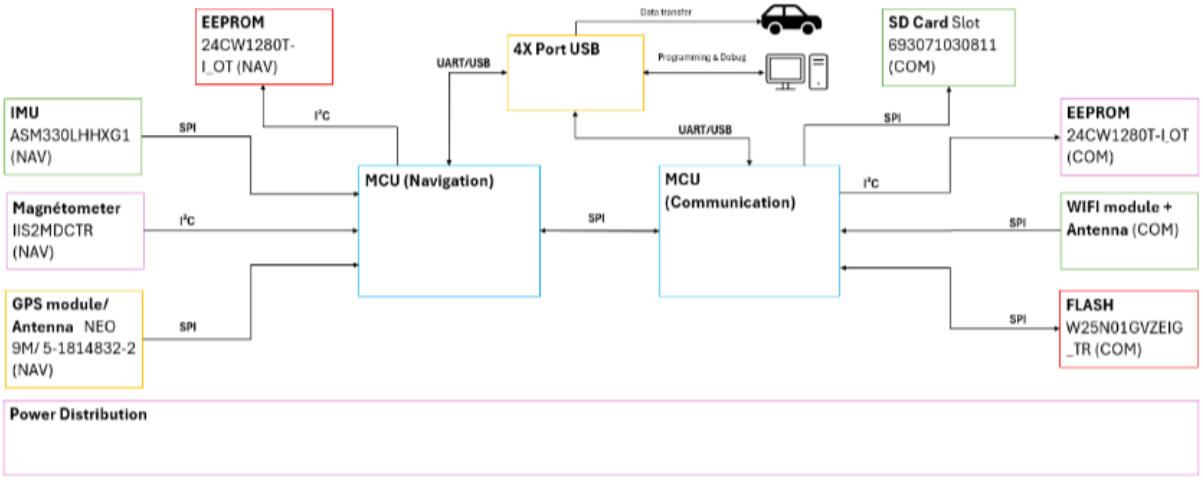

Figure 21 : Schéma bloc de la carte électronique.

Les phases suivantes de CAO ont permis de définir les routages ainsi qu'un modèle final de la carte.

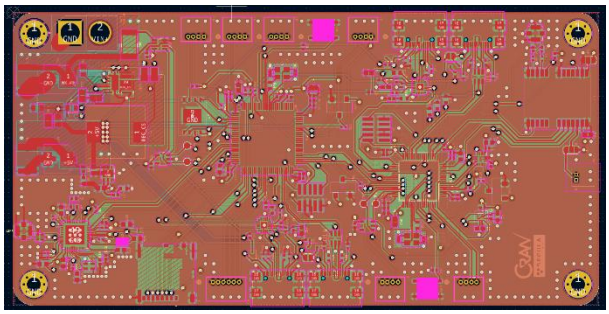

Figure 22 : Image sous KiCad du routage final de la carte embarquée.

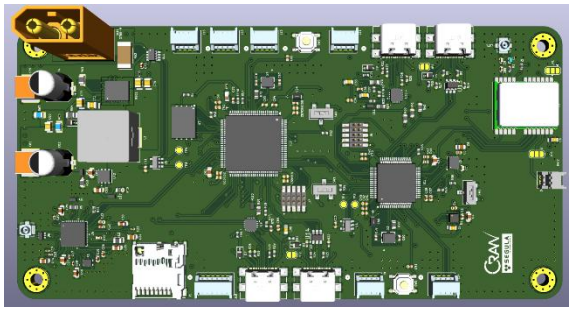

Figure 23 : Image sous KiCad du modèle 3D final de la carte embarquée.

Le projet implique également la réalisation d'une programmation partielle ou complète de la carte embarquée, dans le but de valider le bon fonctionnement du matériel. Nous avons décidé d'utiliser, dans un premier temps, des modules de développement afin de préparer un code fonctionnel en amont. Ces modules ont été exploités sur plaquette de test afin de valider les fonctionnalités et d'anticiper les éventuelles contraintes d'implémentation.

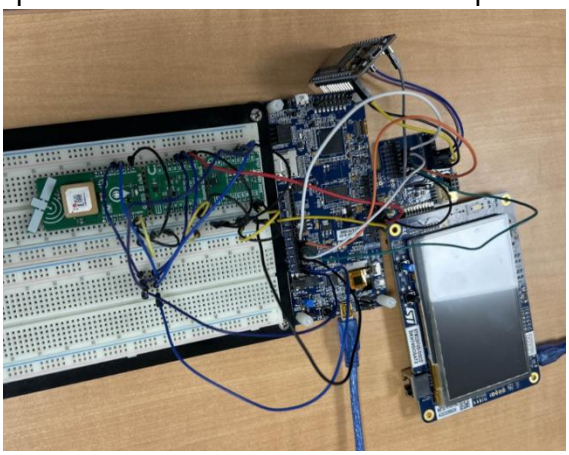

Figure 24 : Photographie du montage expérimental réalisé à l'aide du kit de développement.

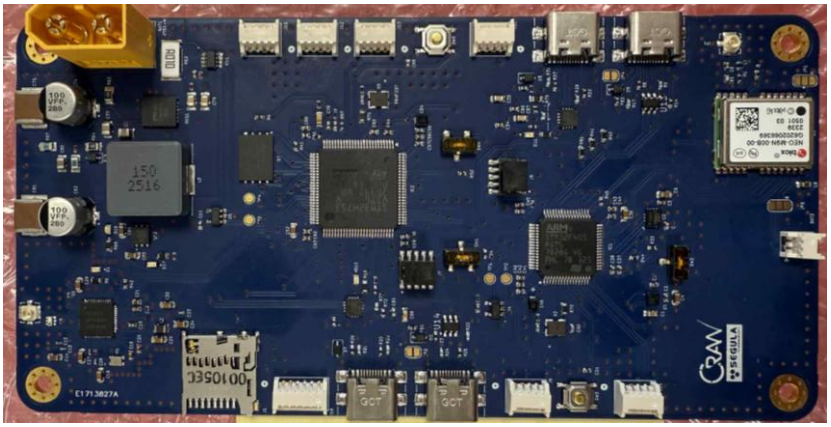

Figure 25 : Image de la carte fabriquée

Nous allons utiliser 3 types de protocoles de communications filaire durant ce projet qui sont SPI, I²C et UART. L'utilisation de ces différents protocoles de communication contribue à atteindre l'objectif principal de la carte, qui est de servir d'observateur pour le véhicule. Elle permettra ainsi de détecter d'éventuelles défaillances et de prévenir les risques liés aux

cyberattaques développé au cours de ce projet. Le code embarqué doit permettre de réaliser les fonctions suivantes :

- Récolter les informations sur l'environnement à l'aide des capteurs embarqués.
- Structurer ces données dans une liste dynamique adaptée au traitement en temps réel.
- Communiquer avec les véhicules voisins afin d'échanger les données environnementales.
- Fusionner les informations reçues avec celles mesurées localement pour obtenir une vision cohérente.
- Transmettre les données consolidées au système du véhicule pour une prise de décision.

Les premiers tests du firmware ont permis de valider :

- La lecture et la transmission des données capteurs via UART ;
- Le fonctionnement stable du scheduler RTOS à 100 Hz ;
- Une gestion correcte des interruptions et des tâches asynchrones ;
- Une compatibilité complète entre les périphériques configurés et le matériel cible.

~~5.4~~5.3 Bilan des travaux

~~5.4.1~~5.3.1 Simulations virtuelles

Au cours de cette phase, l'algorithme initial a été porté de MATLAB vers Python puis entièrement adapté à l'écosystème QLabs. Une architecture logicielle modulaire a été développée, séparant clairement l'environnement, le véhicule, la communication, la synchronisation GPS et la journalisation, ce qui a facilité les tests incrémentaux et la réutilisation du code. La communication V2V a été rendue robuste grâce à l'usage d'UDP avec accusés de réception, « heartbeat » et horodatage, assurant la cohérence temporelle des échanges. Côté commande, le suivi longitudinal s'appuie sur CACC tandis que le suivi latéral intègre un contrôleur ELC afin de limiter le « corner-cutting » en virage. Enfin, la chaîne d'observation comprend un observateur local pour le filtrage et la reconstruction des états et un observateur distribué pour la reconstruction de la flotte via V2V, tandis que la perception multimodale LiDAR-caméra (segmentation, projection et association 2D-3D) fournit d par le contrôle et le module de confiance.

Les expérimentations en scénario nominal montrent une bonne concordance entre les estimations de l'observateur local et le GPS pour (x, y, ψ, v). L'observateur distribué permet à chaque véhicule de reconstruire l'état de l'ensemble de la flotte avec des trajectoires et des vitesses cohérentes. Le contrôleur latéral ELC réduit le « corner-cutting » observé en virage. La fusion LiDAR-caméra génère des estimations fiables de d et v, confirmant l'apport de la perception multimodale pour les boucles de contrôle et d'estimation. Les résultats obtenus confirment la bonne robustesse du système face aux attaques sur la position, la distance inter-véhiculaire restant stable dans tous les scénarios. En revanche, les attaques sur la vitesse provoquent une augmentation notable de l'erreur de vitesse, démontrant une plus grande sensibilité de cette variable aux falsifications. Ces observations valident la pertinence du cadre de confiance intégré à l'observateur distribué pour renforcer la résilience du système coopératif.

~~5.4.2~~5.3.2 Préparation aux essais en environnement réel

Ces travaux ont permis de développer une carte électronique embarquée dédiée à l'estimation avancée et à la détection des cyberattaques dans les véhicules autonomes et connectés. Les principaux résultats peuvent être résumés ainsi :

- La carte électronique conçue constitue une base matérielle robuste, dont la validation sera confirmée par les tests fonctionnels.
- Le choix des composants et la définition du *stackup* PCB ont permis d'assurer une compatibilité électromagnétique satisfaisante et une adaptation correcte des impédances pour les lignes de communication sans fil, malgré certains points d'optimisation identifiés.
- Le développement a mis en évidence plusieurs axes d'amélioration, notamment l'optimisation du réseau d'adaptation RF, la réduction des pertes liées aux vias, l'amélioration du routage haute fréquence et la gestion plus fine des alimentations.
- Sur le plan logiciel, la programmation embarquée est encore limitée mais constitue une étape essentielle pour valider le bon fonctionnement du matériel et préparer un environnement adapté aux futures évolutions.

6 Ressources humaines

Cf. documents économiques

7 Partenariat scientifique et recherche confiée

Ce projet s'inscrit pour partie dans une collaboration avec le Centre de Recherche en Automatique de Nancy (CRAN) et bénéficie d'une thèse CIFRE de NGUYEN Quang Huy. Le laboratoire apporte son expertise dans les domaines de l'électronique et le développement d'observateurs et contrôleurs. Il met à disposition aussi sa plateforme de test pour la validation des modèles.