

A custom-built mobile robot is shown on a grassy field. The robot has a blue frame, large black treaded tires, and various sensors and electronics mounted on top. A camera or sensor is mounted on a vertical pole. The background shows a modern building with large windows and a clear blue sky.

UENAV : Unstructured Environment NAVigation

Antoine MORVAN, Simon PHILIBERT, Titouan LEOST, Romain BORNIER



ENSTA
BRETAGNE

Introduction

Objectif : Intégration d'une boîte de capteur permettant la téléopération en environnement non structuré



Fig1: Quad désossé sans robotisation



Fig2: 1er prototype d'un quad robotisé et téléopérable à vue

Plan

I - Amélioration de la robotisation

- Système de frein
- Système de direction

II - Intégration des capteurs

- Kinect
- LiDAR
- GNSS et IMU
- Schéma global
- Boîtier électronique mobile et modulable

III - Téléopération

- Pilotage
- Modèle cinématique
- IHM

IV - Tests expérimentaux

- Téléopération par IHM
- SLAM

V – Perspectives

Conclusion

I - Amélioration de la robotisation

Système de frein

/ Limites de la première version développée en 2^{ème} année:

- Imprécis car peu de débattement
- Peu efficace
- Peu robuste

Adapté à un premier prototype mais pas à une plateforme fonctionnelle

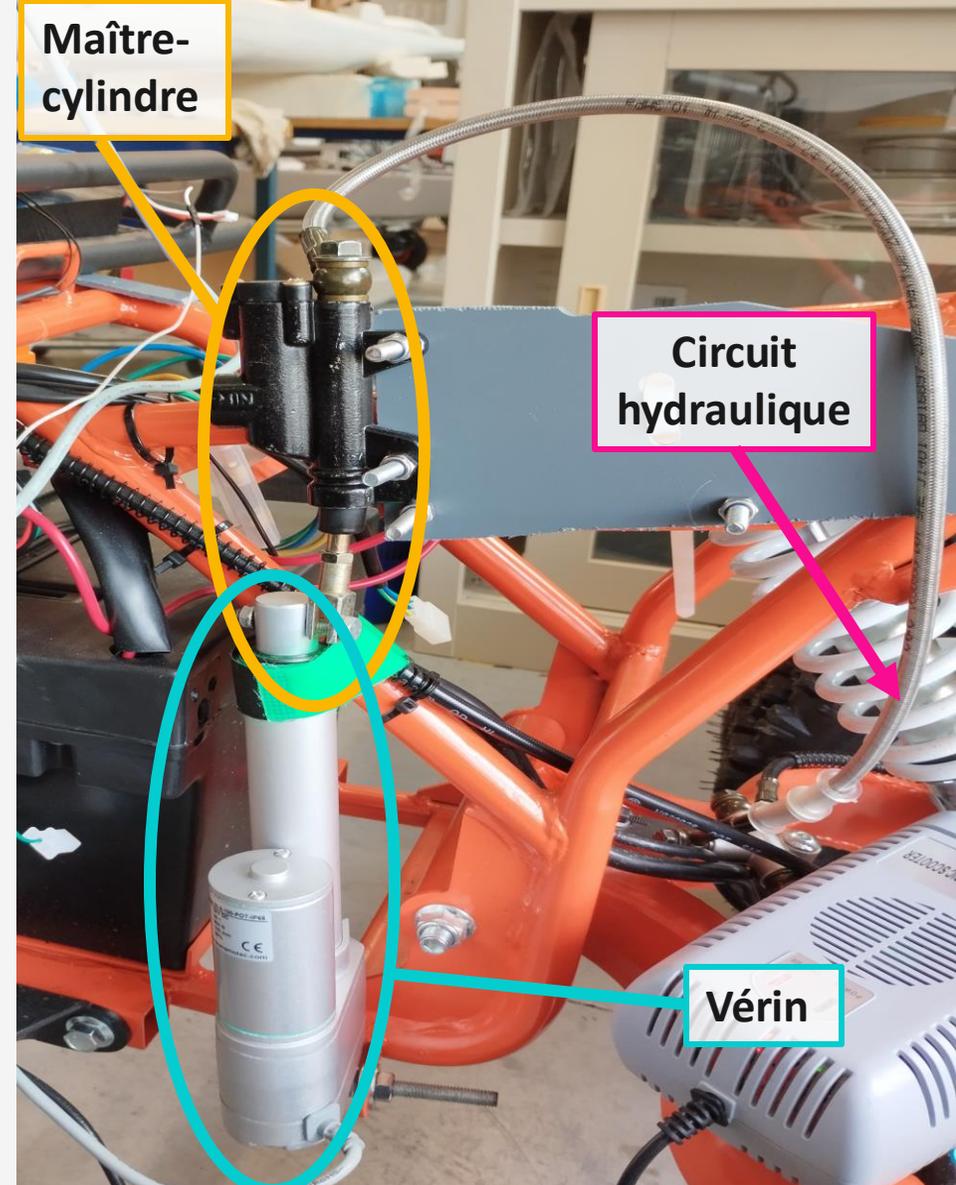


Fig3: Montage du premier système de frein

Système de frein

- / Nouveau système de frein :
 - **Robuste** : fixation sur cadre fait en profilés
 - **Précision et vitesse réglables**

Évolution de la distance de freinage en fonction du système de frein

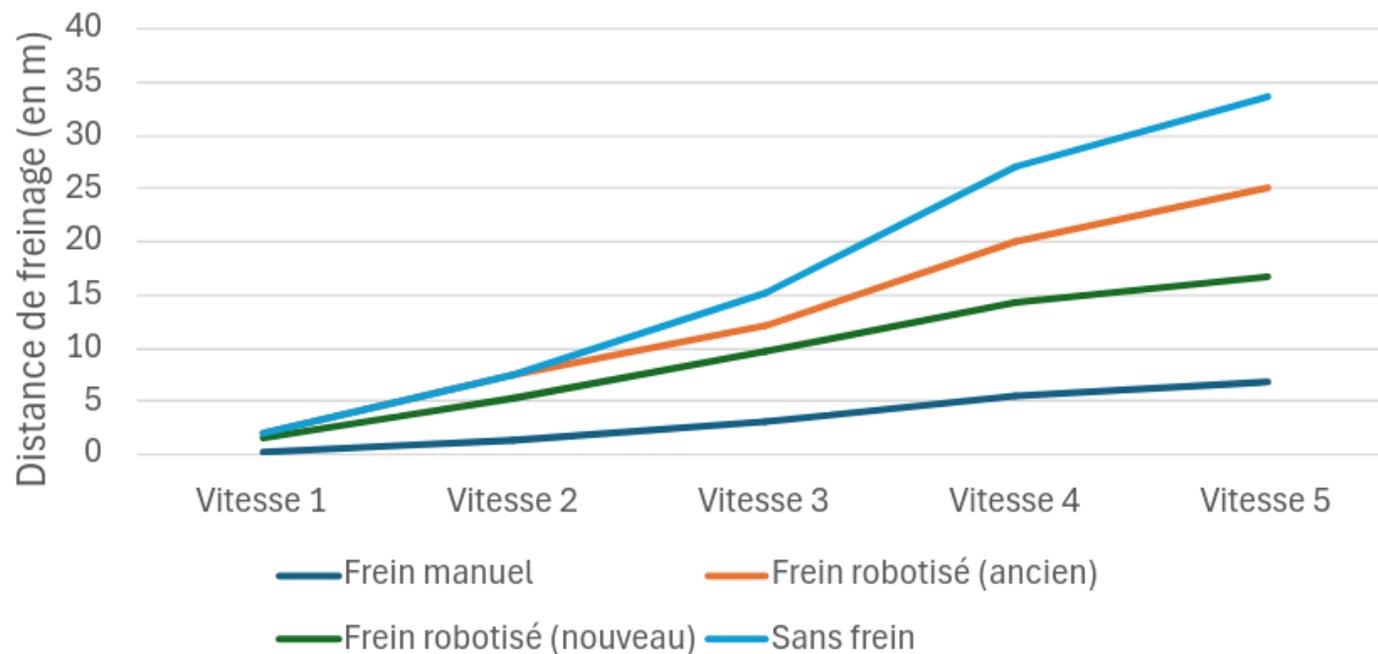


Fig4: Montage du nouveau système de frein

Système de direction

/ Première version:

- Vérin précis au millimètre mais très lent (10 secondes de butée à butée)
- Support peu robuste et déformable

Permettait une première prise en main du système de direction, mais la lenteur du système le rendait inutilisable pour de la téléopération en milieu déstructuré

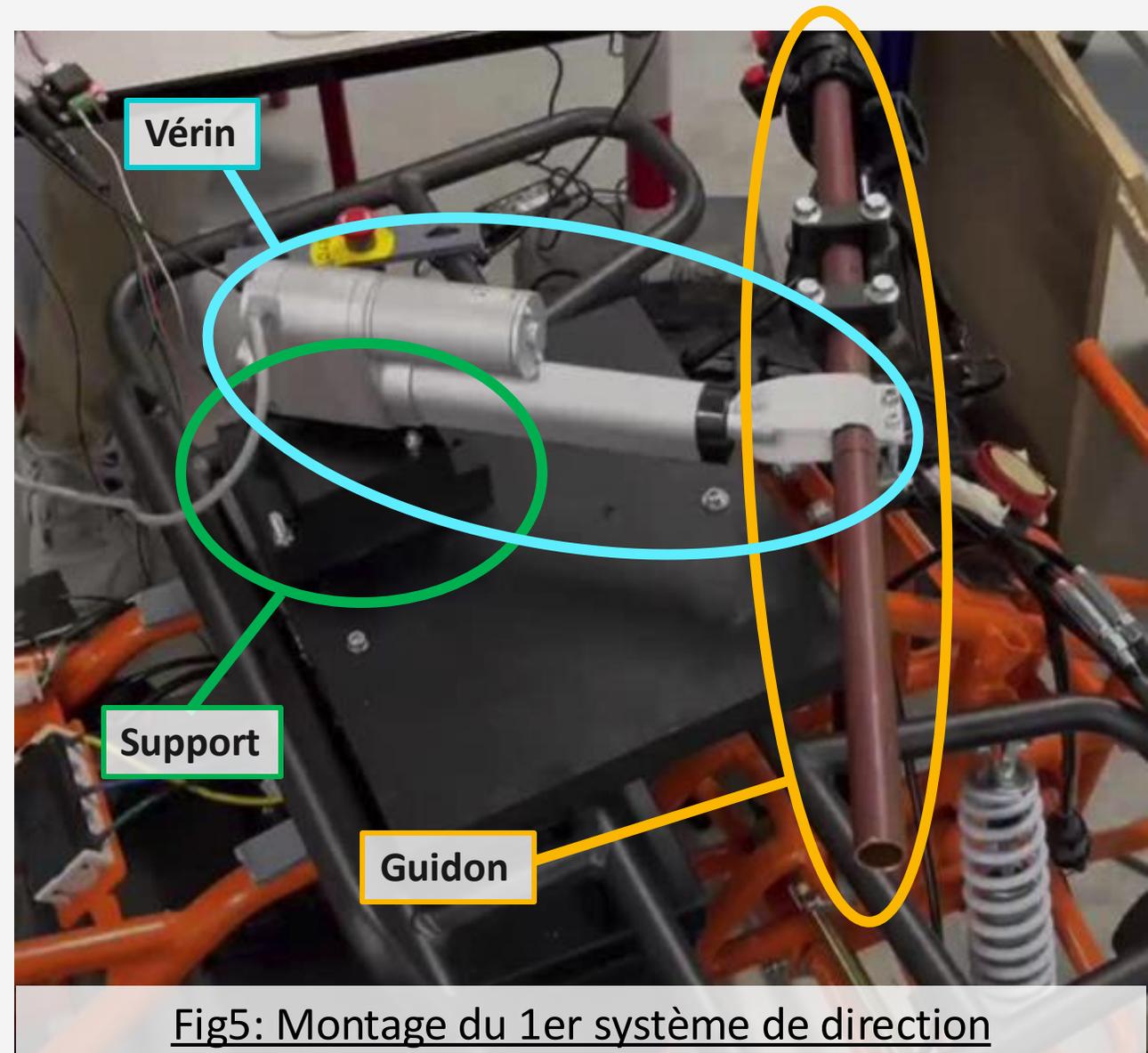


Fig5: Montage du 1er système de direction

Système de direction

/ Deuxième version:

- Vérin précis au millimètre et très rapide (2 secondes de butée à butée)
- Support solide et facilement ajustable
- Vérin plus puissant (Force de 1700N)
- Accès à plus d'informations et paramétrable (courant, position, rampe, etc.)

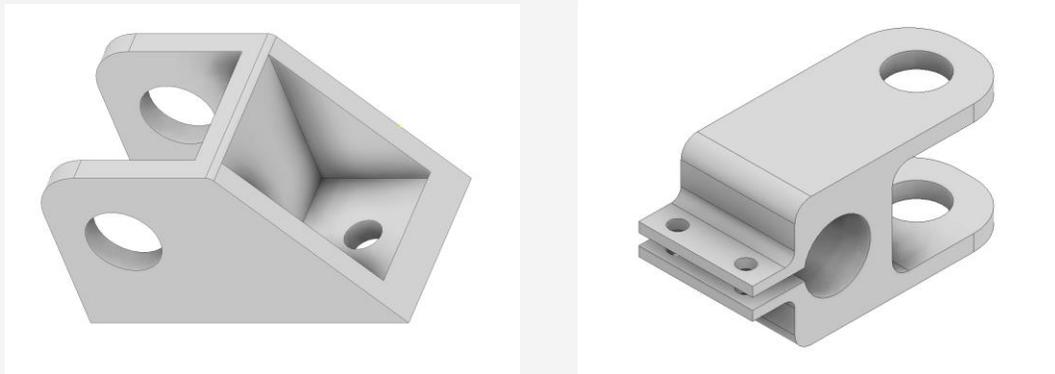


Fig6: CAO des nouvelles pièces d'adaptation: Vérin - Quad

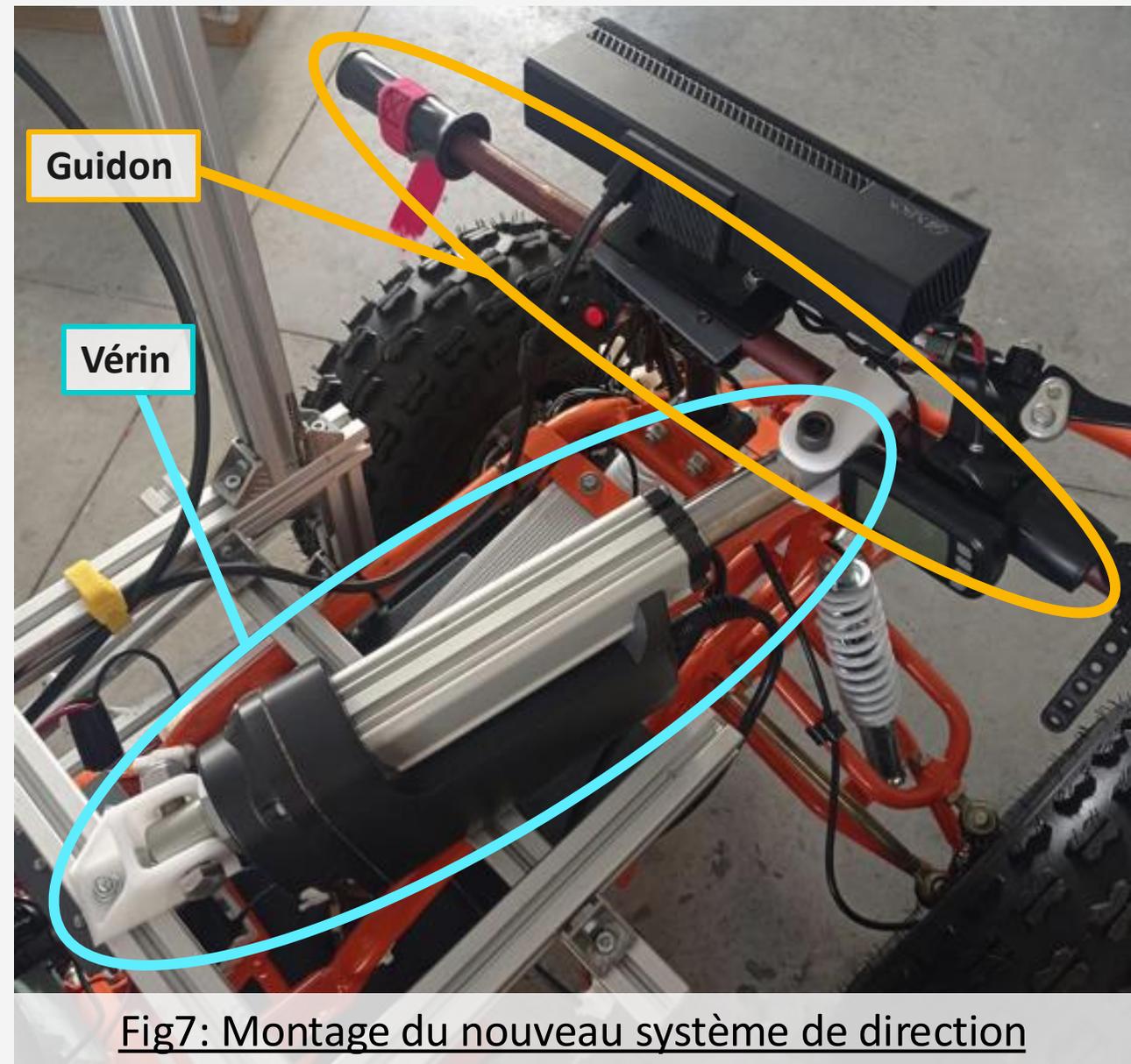


Fig7: Montage du nouveau système de direction

Essais des 2 systèmes de direction

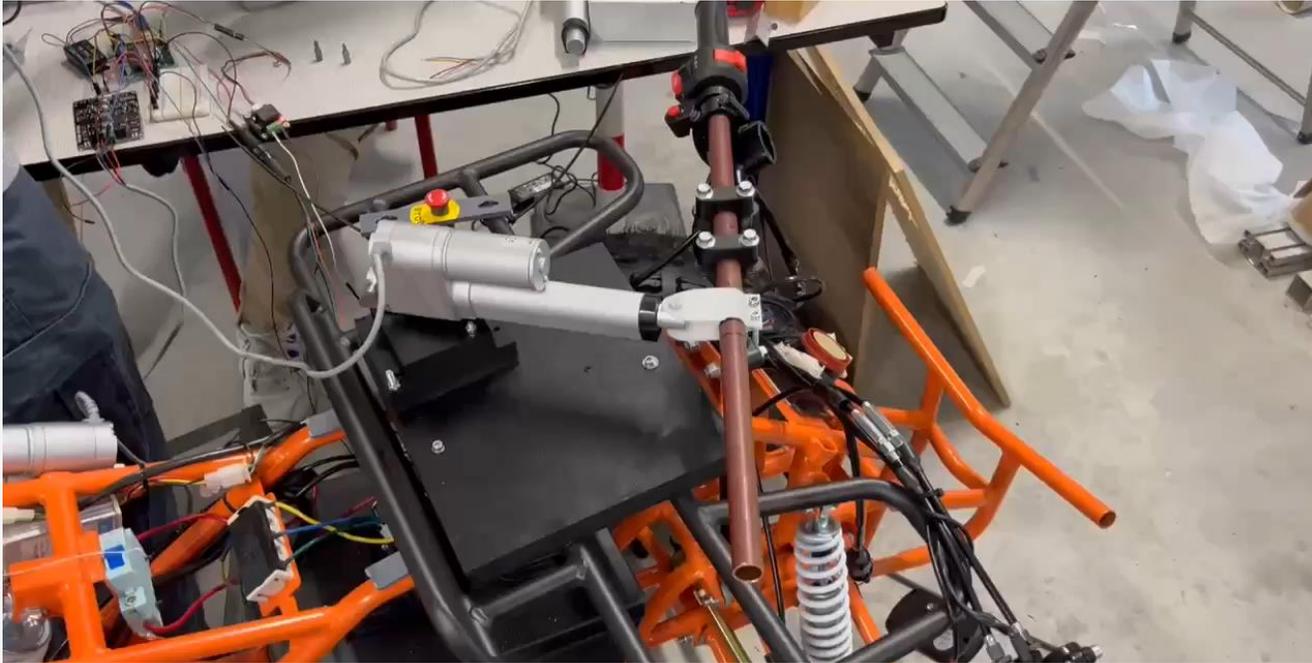


Fig8: Vidéo de fonctionnement de butée à butée de l'ancien vérin

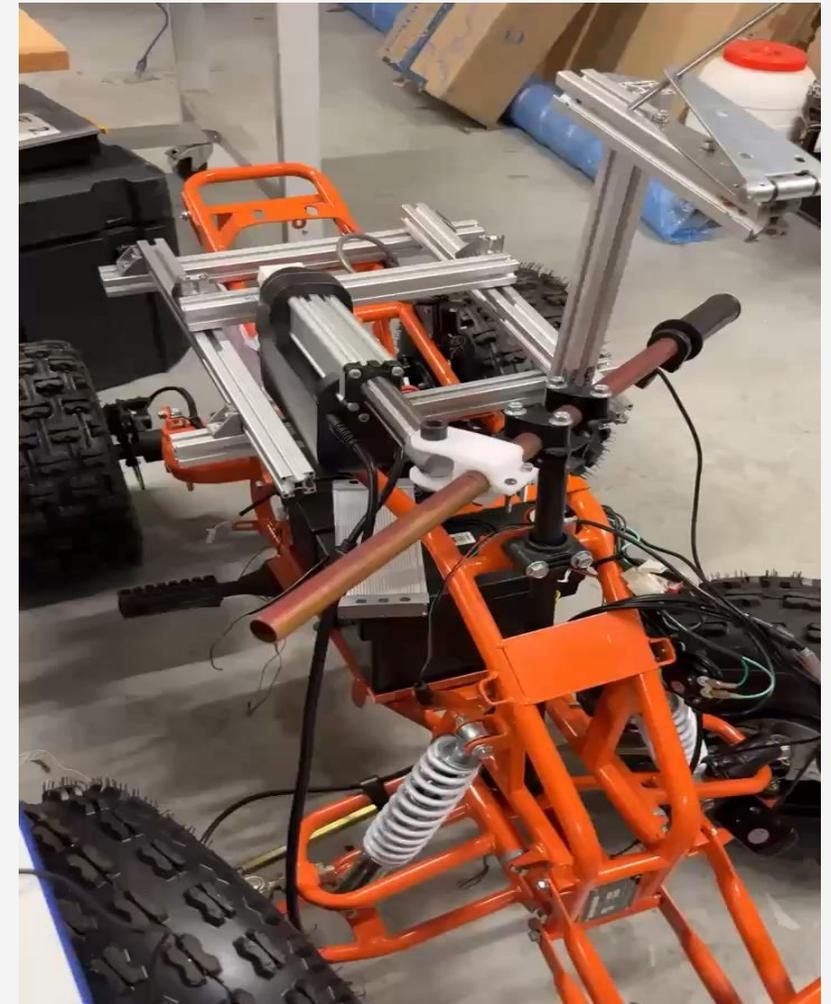


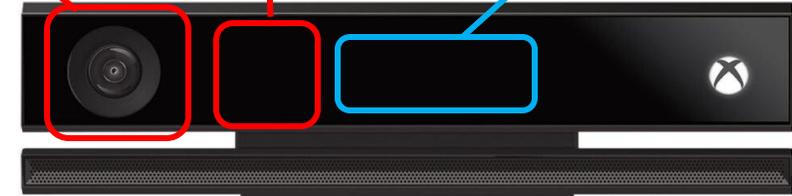
Fig9: Vidéo de fonctionnement de butée à butée du vérin LINAK

II - Intégration des capteurs



Fig10: Montage de la caméra Kinect

Capteur RGB Capteur IR Émetteur IR



- / Permet au pilote de voir devant le quad
- / Intégration simple en s'appuyant sur l'api *Libfreenect2*
- / Mise en place d'un nœud ROS -> **Limitation du débit**
 - Compression JPEG
 - Redimensionnement de l'image
 - Passage en monochrome
 - Fréquence de diffusion

LiDAR

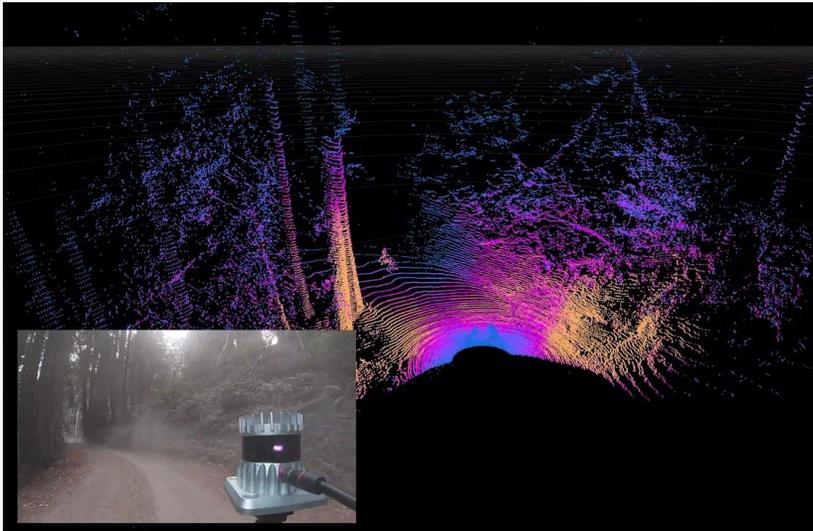


Fig11: LiDAR Ouster et nuage de points

Objectif :

Connaissance 3D de l'environnement

- Exploitation du **SLAM** : (*Simultaneous Localisation And Mapping*)
- **Détection d'obstacles** et planification de **trajectoire**
- Estimations de **franchissabilité**

Monté sur support orientable pour visibilité avant

Caractéristiques :

- Gyroscope et accéléromètre
- Profondeur
- Réflectivité
- Proche infrarouge

Champ de vision : 360 ° (h) 90 ° (v)

Résolution : 0,01 °

Portée : 30 cm à 45 m

Précision (profondeur) : ~1 cm

Fréquence : 10 Hz / 20 Hz

GNSS

/ GlobalSat G-STAR IV

- Permet de localiser le véhicule
- Mesure de code (précision 1-3 m)
- IP67



Fig12: Montage du GNSS

Centrale inertielle (IMU)

/ SBG Ellipse A

- Gyroscope, magnétomètre et accéléromètre
- Permet d'avoir l'attitude du robot (roulis, tangage, lacet)

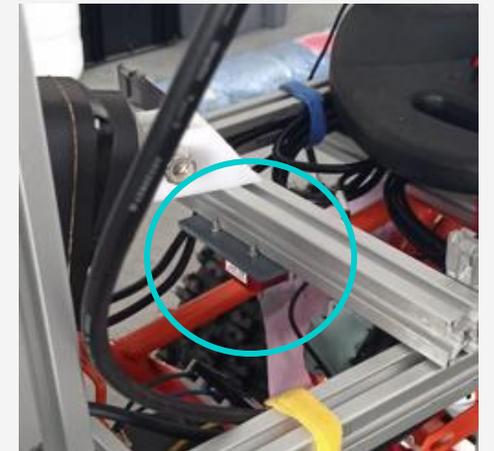


Fig13: IMU et montage de l'IMU

Boîtier électronique mobile et modulable

/ Boîtier regroupant l'électronique et les interfaces des capteurs, facilitant le déploiement rapide de la configuration des capteurs sur un autre véhicule robotisé.

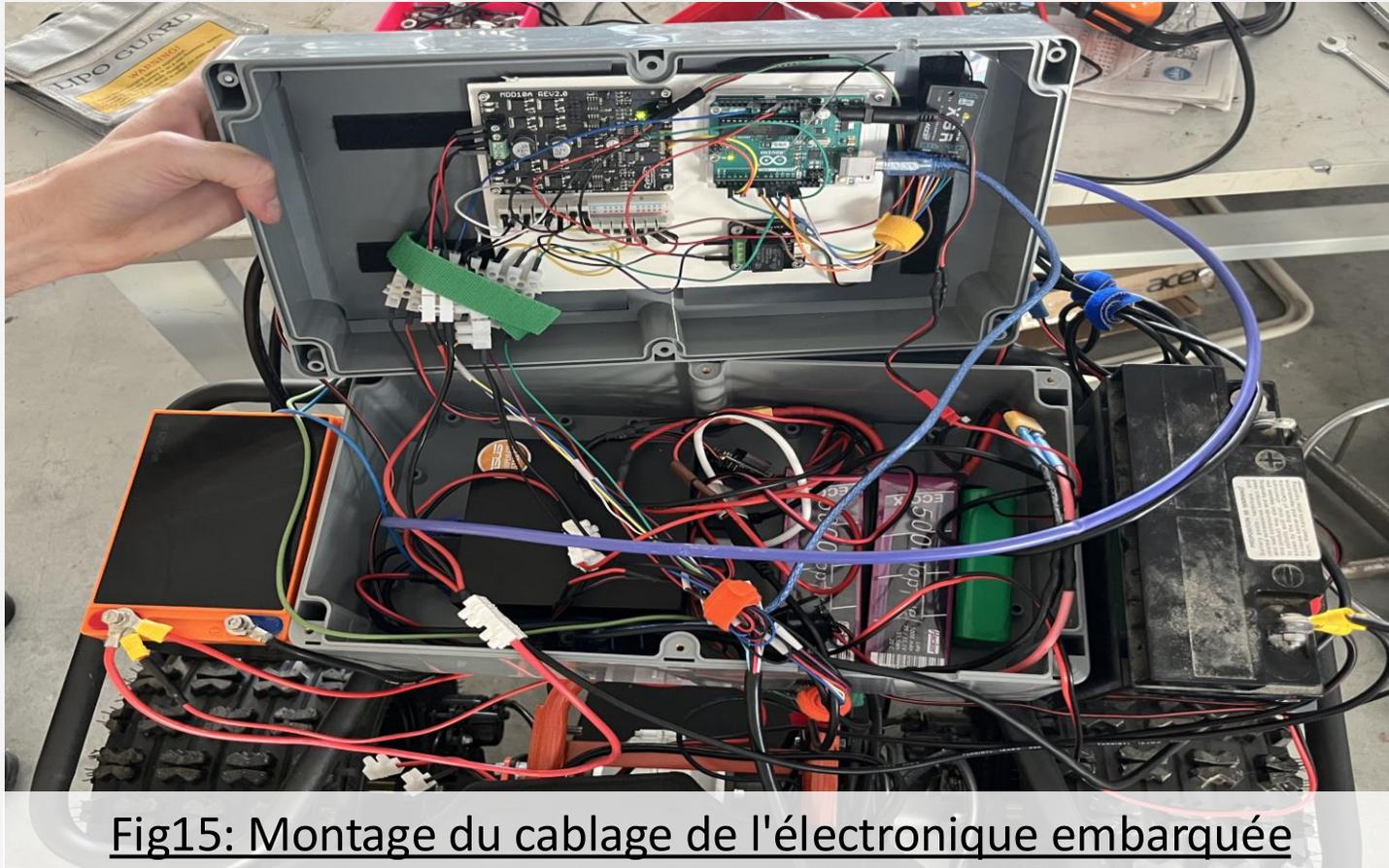


Fig15: Montage du cablage de l'électronique embarquée

Consommation / autonomie :

- PC Asus : 15 W -> 3 h 40 min
 - Kinect : 12 - 21 W -> 3 h
 - Vérin de direction : 150W
 - LiDAR : 17 W
 - Vérin de freinage : faible
- } 1h40
- IMU : 0,25 W
 - GPS : 0,28 W

III - Téléopération

Pilotage

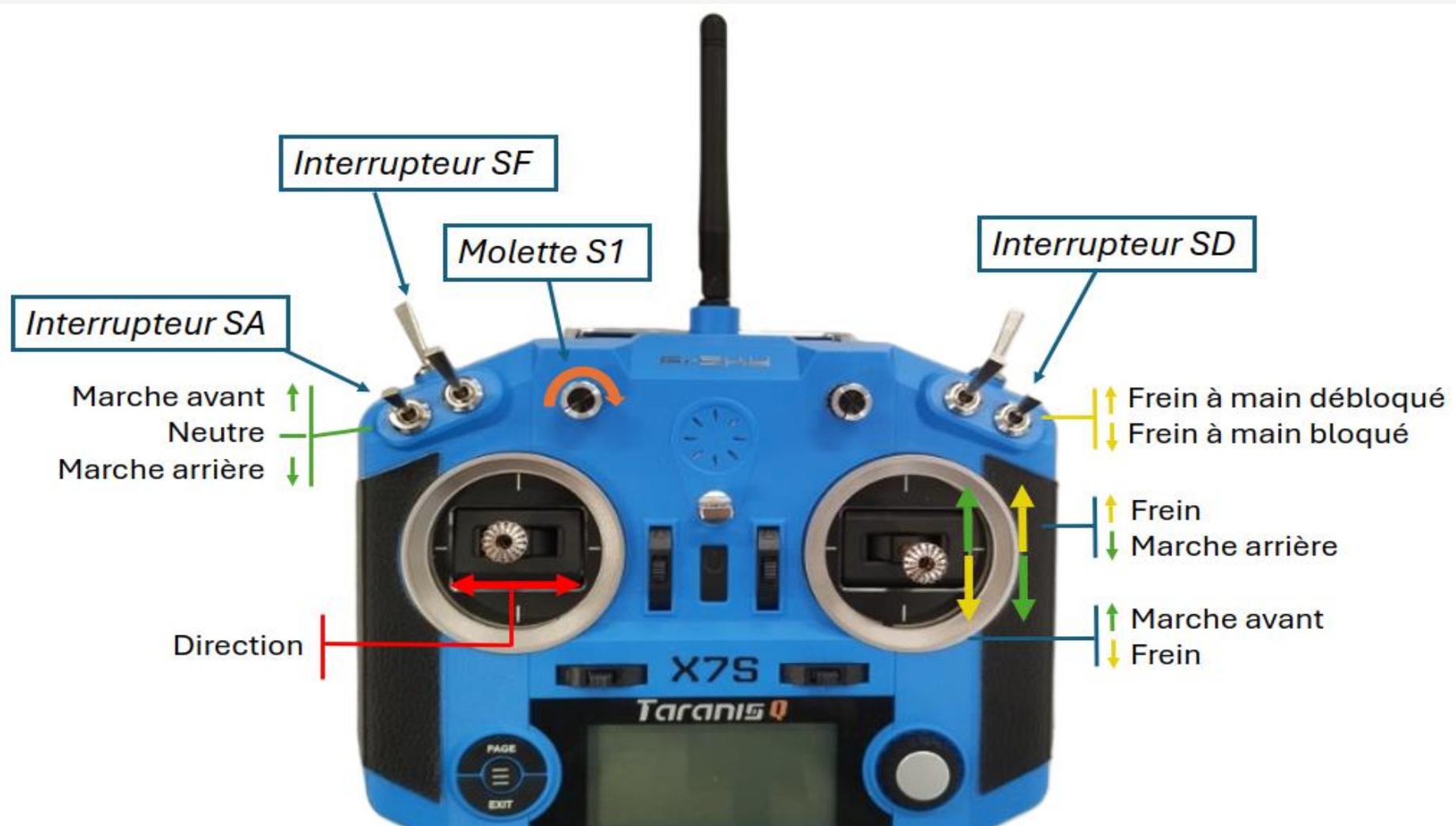


Fig16: Schéma d'utilisation de la télécommande

Modèle cinématique

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix}}_{\dot{\eta}} = \underbrace{\begin{bmatrix} \cos \psi & 0 \\ \sin \psi & 0 \\ 0 & 1 \end{bmatrix}}_{J} \cdot \underbrace{\begin{bmatrix} u \\ r \end{bmatrix}}_{v} \rightarrow \dot{\eta} = J \cdot v$$

u : commande frein
δ : commande direction
f : commande frein
L : distance entre les roues

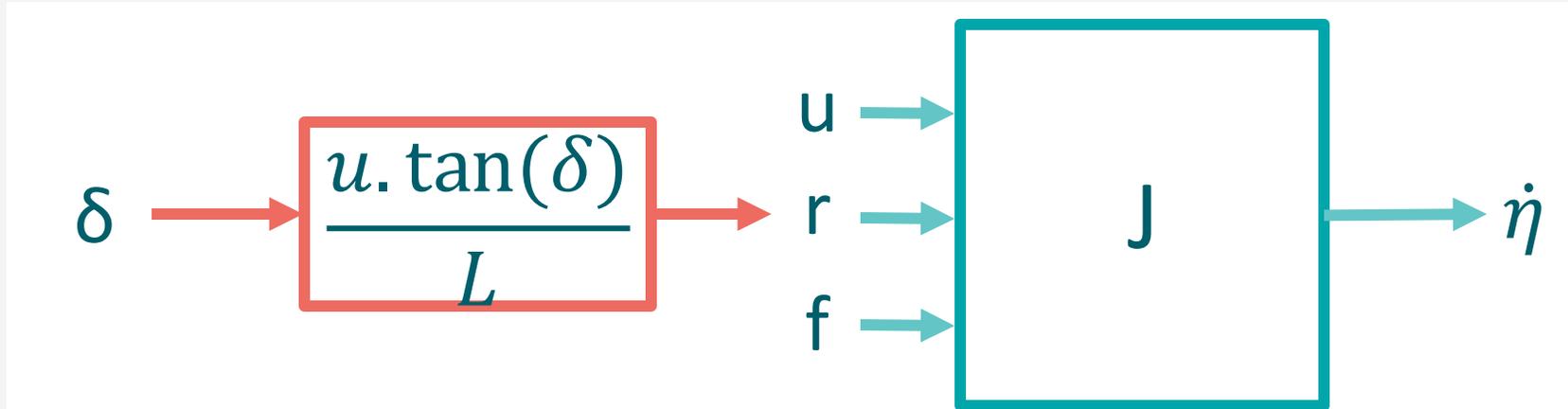


Fig17: Modèle cinématique du quad robotisé

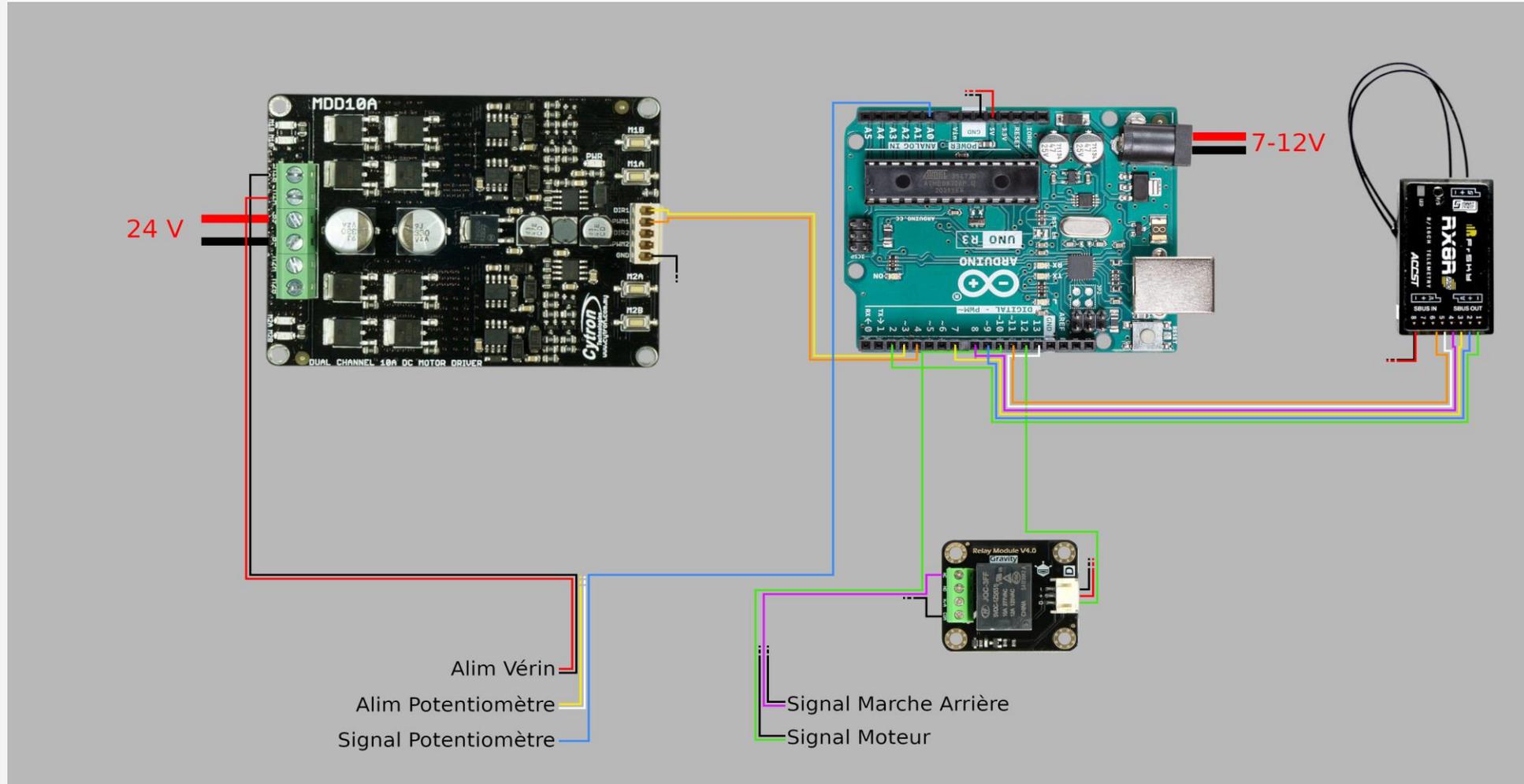


Fig18: Schéma électronique pour la téléopération

/ IHM générée sur python avec le module PYQT5

Formules: Quaternions -> Angles d'Euler

$$\phi = \arctan 2(2(q_w q_x + q_y q_z), 1 - 2(q_x^2 + q_y^2))$$

$$\theta = \arcsin(2(q_w q_y - q_z q_x))$$

$$\psi = \arctan 2(2(q_w q_z + q_x q_y), 1 - 2(q_y^2 + q_z^2))$$

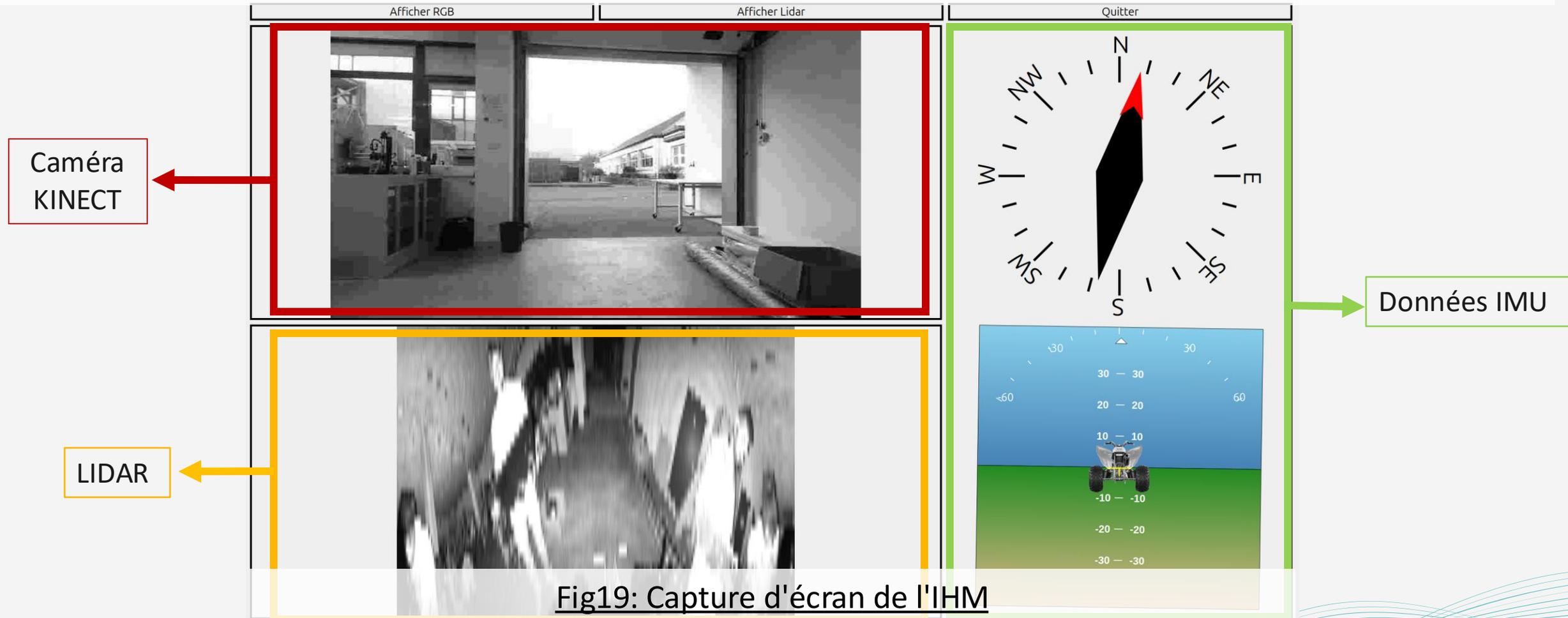


Fig19: Capture d'écran de l'IHM

IV - Tests expérimentaux

Téléopération par IHM (Vue réelle)

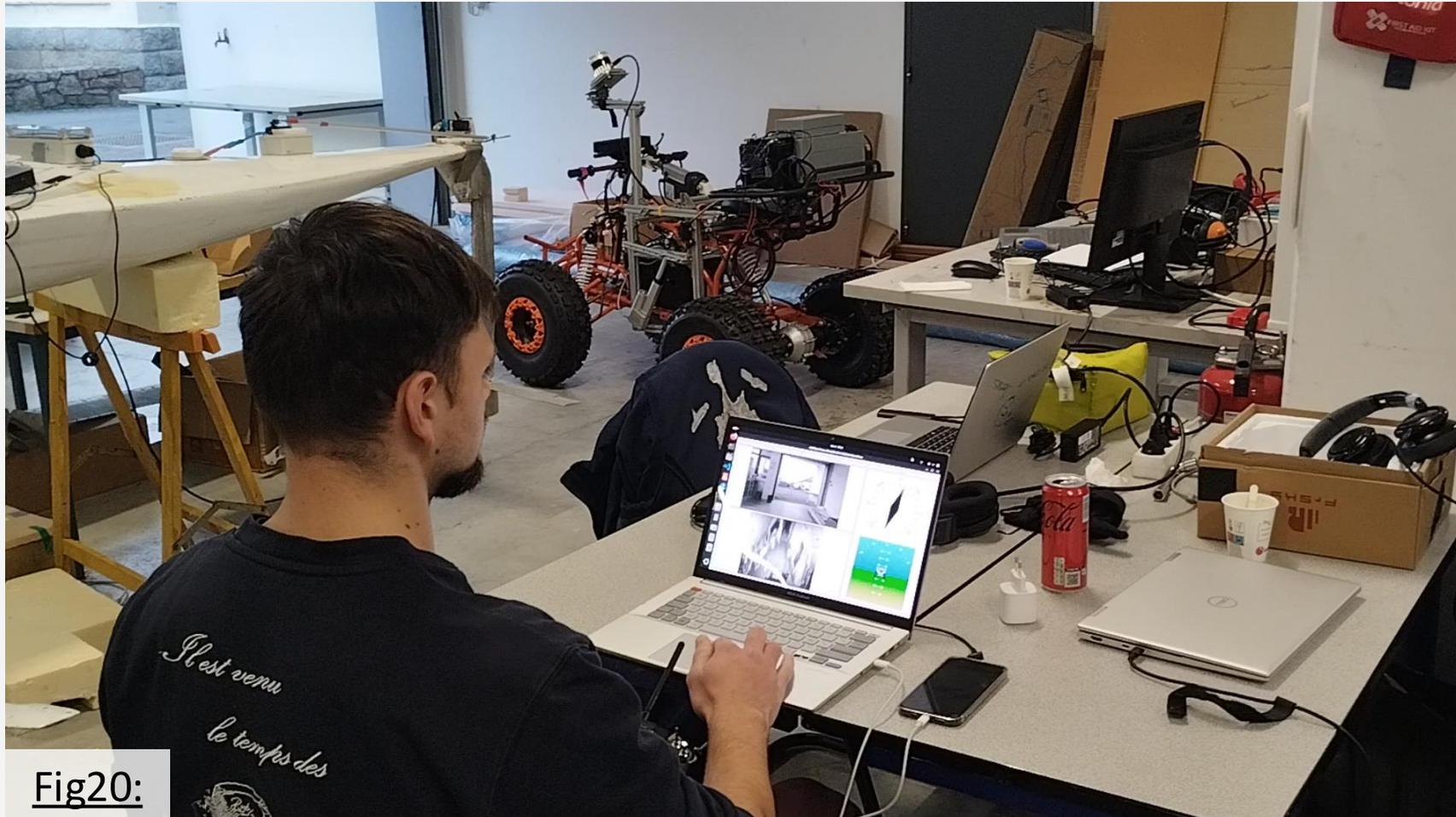


Fig20:

Téléopération par IHM (vue IHM)

The screenshot displays a software interface for remote operation, divided into several panels:

- Recording Control Panel:** Features a "Start recording" button with a red indicator. Below it are options for "Schedule: (inactive)", "Activate schedule", and "Edit schedule". Checkboxes for "Enable recording hotkey" (checked) and "Enable sound notifications" are present. A "Hotkey" dropdown menu is set to "Super + R".
- Information Panel:** Lists recording statistics: "Total time: 0:00:00", "FPS in: 0.00", "FPS out: 0.00", "Size in: 2016x1225", "Size out: ?", "File name: ?", "File size: 0 B", and "Bit rate: 0 bit/s".
- Preview Panel:** Includes a "Preview frame rate" dropdown set to "10" and a note: "Note: Previewing requires extra CPU time (especially at high frame rates)". A "Start preview" button is located at the bottom.
- Log Panel:** Contains a text area with entries: "[PageRecord::StartPage] Starting page ..." and "[PageRecord::StartPage] Started page."
- Bottom Controls:** Includes "Cancel recording" and "Save recording" buttons.
- Video Feed:** A window titled "Afficher Lidar" shows a live camera view of a robot's environment.
- Compass:** A window titled "Quitter" displays a compass rose with cardinal and ordinal directions (N, NE, E, SE, S, SW, W, NW).
- 3D Sensor Visualization:** A window showing a top-down view of a robot on a green field. A grid of numerical values is overlaid on the field, ranging from -60 to 30, representing sensor data.

Fig21:

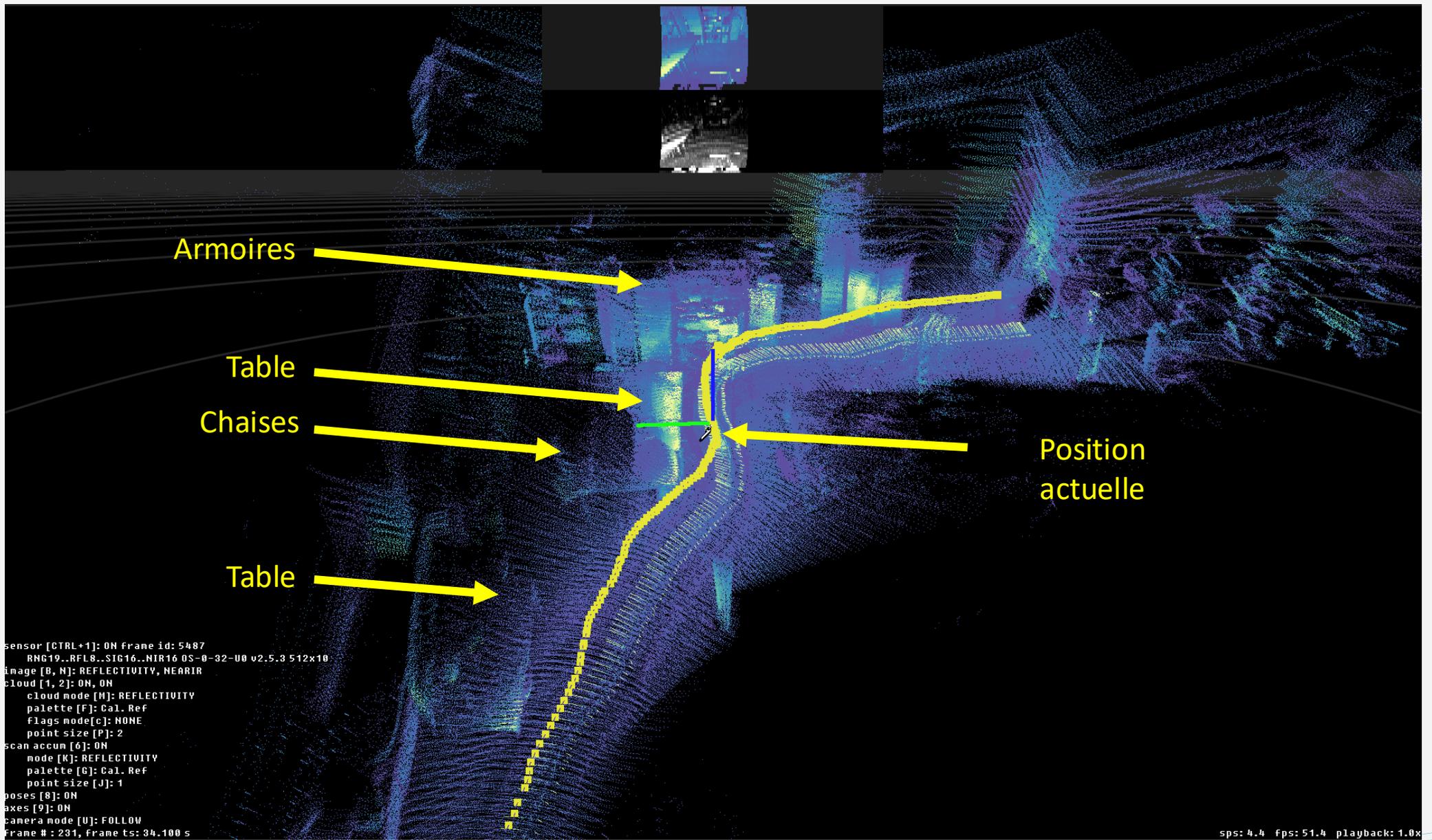


Fig22: Résultat d'un essai d'acquisition SLAM en intérieur (Labo de robotique)

V - Perspectives

Améliorations :

- Sécurité (arrêt d'urgence)
- Circuit imprimé custom
- Réseau

Développement du projet :

- Documentation complète réalisée
- Plateforme adaptée au développement d'algorithme d'autonomie

Conclusion

/ Passage d'un 1er prototype à un quad entièrement robotisé, fonctionnel et téléopérable sous capteurs



Fig23: Photographie du quad dans sa version finale

Questions

Annexe

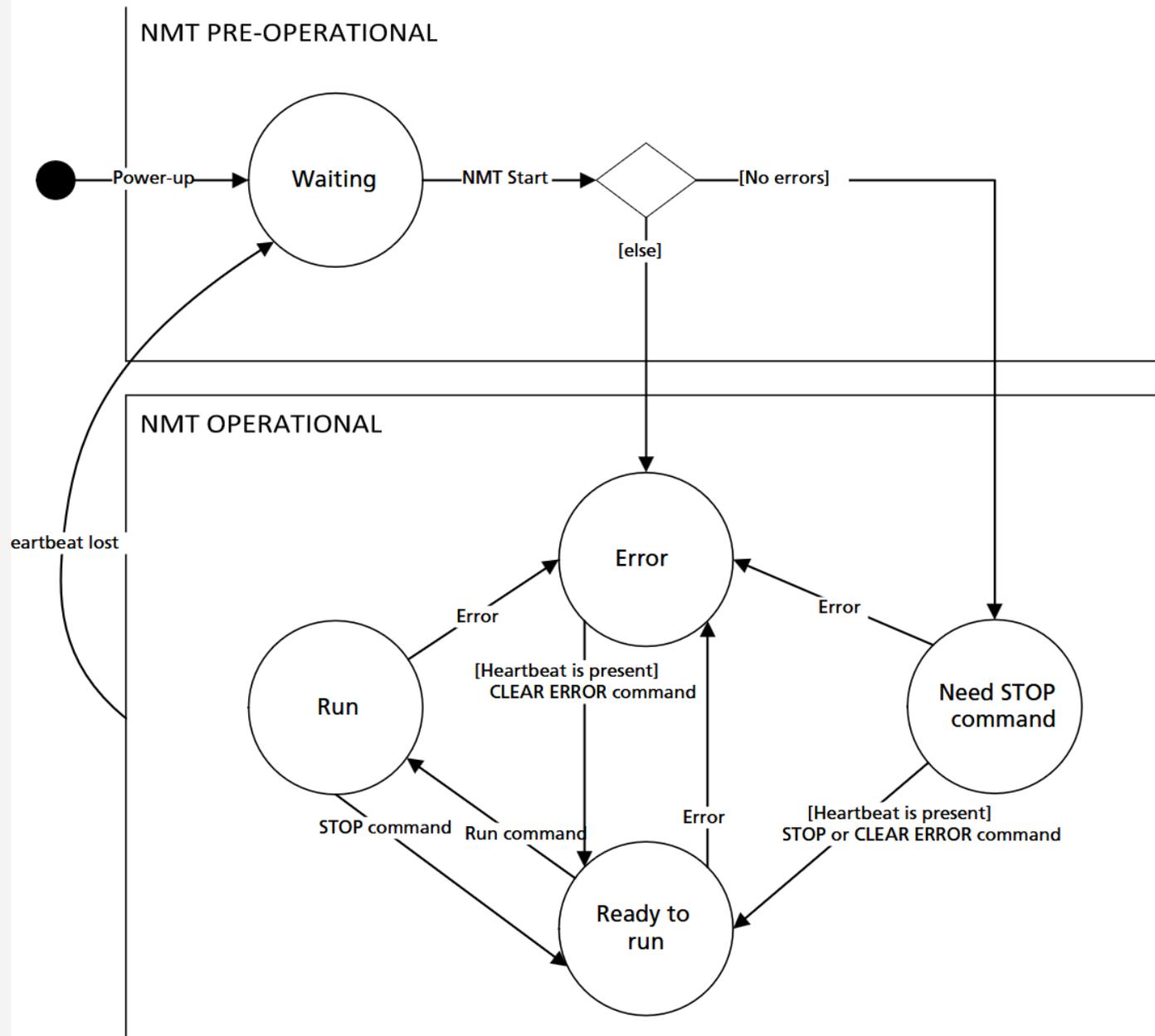


Schéma de démarrage du vérin Linak

Starting procedures

Follow the example below to complete the startup procedures necessary for successful communication.

The CAN address in this example is in 32 (0X20).

CAN ID [hex]	Data [hex]								Description	
	0	1	2	3	4	5	6	7		
701	05								Master heartbeat. Sent every 100 ms	Startup Procedures
720	00								Actuator boot-up	
620	23	16	10	01	C8	00	01	00	Receive Service Data Objects (SDO)	
5A0	60	16	10	01	00	00	00	00	Actuator response	
000	01	20							Network Management (NMT) Start	
1A0	00	00	00	C1	01	00	00	C0	Transmit Process Data Object TPD01. 'Stop' command needed.	
220	03	FB	FB	FB	FB	FB	00	00	'Stop' command.	
1A0	00	00	00	C1	00	00	00	C0	Transmit PD01. Actuator at Endstop reached in [0 mm]	Running Examples
220	01	FB	FB	FB	FB	FB	00	00	'Run out' command	
1A0	A1	00	06	C8	00	31	00	C0	Transmit PD01. Actuator is running out [2.6 mm]	
1A0	F7	01	00	C2	00	00	00	C0	Transmit PD01. Actuator reached endstop outwards [50.3 mm]	
220	02	FB	FB	FB	FB	FB	00	00	'Run in' command	
1A0	60	00	06	D0	00	32	00	C0	Transmit PD01. Actuator is running in [9.6 mm]	
1A0	00	00	00	C1	00	00	00	C0	Transmit PD01. Actuator reached endstop inwards [0 mm]	

Procédure de démarrage du vérin Linak

Process Data Objects (PDO)

RPD01 is mapped to 0x2000

TPD01 is mapped to 0x2001

Command details

Index	Subindex	Command	Data type	Details	Description	Unit
0x2000	1	Position	UINT16	0-64255 64256 64257 64258 64259 64260 64261 64262-65535	Run to position Clear ErrorCode register (see 0x1001) Command run actuator out Command run actuator in Command stop actuator Command run actuator out (Recovery mode) Command run actuator in (Recovery mode) Invalid value, actuator will not run	0.1 mm/ bit
	2	Current	UINT8	0-250 251 252-255	Max. current limit Use default current value Invalid value, actuator will not run	0.25 A/bit
	3	Speed	UINT8	0-200 201-250 251 252-255	Speed to use Use 100% speed Actuator default speed value Invalid value, actuator will not run	0.5% /bit
	4	Soft Start	UINT8	0-250 251 252-255	Start ramping time (ms) Use default soft start value Invalid value, actuator will not run	0.05 s/bit
	5	Soft Stop	UINT8	0-250 251 252-255	Stop ramping time (ms) Use default soft stop value Invalid value, actuator will not run	0.05 s/bit

Commandes de
contrôles du vérin

Annexe

Feedback status details

Index	Subindex	Command	Data type	Details	Description	Unit
0x2001	1	Position	UINT16	0-64255 64256-65023 65024 65025-65535	Position of actuator piston Reserved Position lost Reserved	0.1 mm/bit
	2	Current	UINT8	0 1-250 251-253 254 255	Not running Measured motor current Reserved Fault in current measurement circuit Reserved	0.25 A/bit
	3	Status Flags	UINT8	b0 b1 b2 b3 b4 b5 b6-b7	Endstop reached in Endstop reached out Overcurrent Running out Running in CANopen heartbeat needed Reserved	8-bit independent status bit-indicators
	4	Error Codes	UINT8	0 1 2 3 4 5 6 7 8 9 10 254 255	No error Need 'Stop' command Hall error Overvoltage Undervoltage Failed to maintain heartbeat Endstop reached error Temperature error Heartbeat error (internal) SMPS error (internal) Current measurement (internal) Internal fault (not specified) External fault (not specified)	8-bit error code indicating the currently active error of highest priority
	5	Speed	UINT16	0-4015 4016-65535	Speed of actuator piston Reserved	0.1 mm/s / bit
	6	AUX Input	UINT8	b0-b1 b2-b3 b4-b5 b6-b7	Input 1 level Input 2 level Input 3 level Reserved (always1)	25% VCC/ bit

FigX: Descriptif de la trame de feedback envoyé par le vérin

UTILISATION

☰ Getting started

Vérifications préliminaires

Branchements

Connexion au PC embarqué

Lancement des nodes ROS et démarrage de la téléopération

Démarrage du QUAD

Vérification du bon fonctionnement

Utilisation manuelle

Troubleshooting

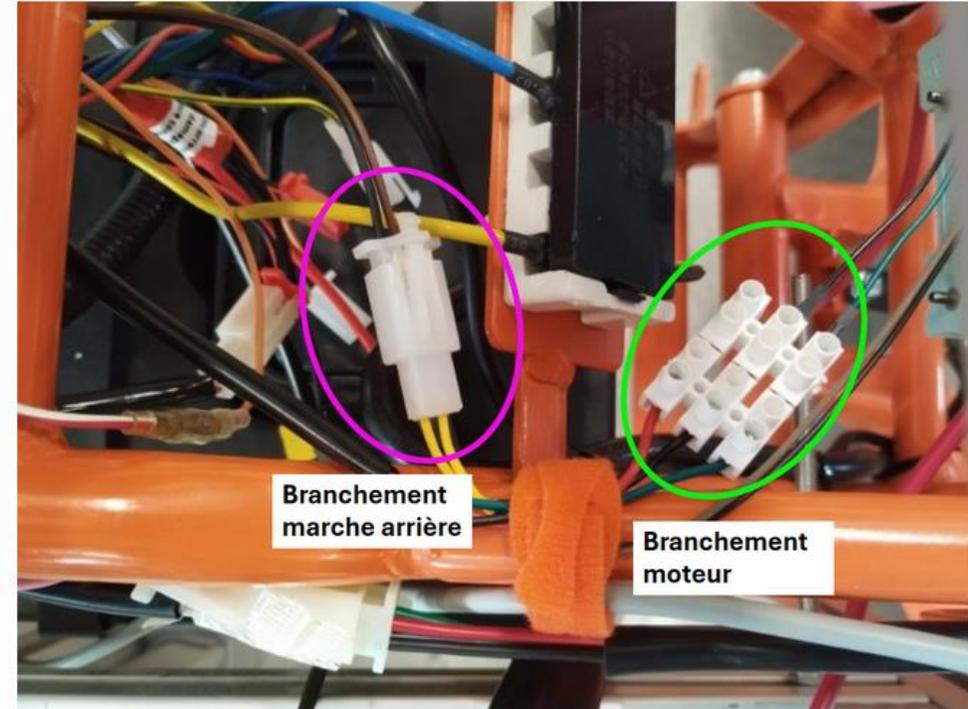
Téléopération

Maintenance

DOCUMENTATION

Teleop package

Kinect package



Branchements du moteur et de la marche arrière

Connexion au PC embarqué

TO DO (connexion au pc en ssh, lancer les nodes, etc.)

Une fois tout les branchements réalisés, connecter un pc de contrôle au pc embarqué via ssh :

1. Ping le pc embarqué afin d'obtenir son adresse ip

```
ping quadarquus-MINIPC-PN50
```

Search docs

UTILISATION

Getting started

Téléopération

Maintenance

DOCUMENTATION

Teleop package

📁 Kinect package

Introduction

⊕ Class *Kinect*

⊕ Class *NodeKinectPub*

Class *NodeKinectPub*

```
class NodeKinectPub : public rclcpp::Node
```

Une classe pour définir le noeud ROS2 de publication des images de la [Kinect](#).

Cette classe permet de publier les images RGB et IR de la [Kinect](#) en fonction de paramètres définis par défaut ou lors de l'appel au launch associé.

Public Functions

```
NodeKinectPub()
```

Constructeur de la classe.

Le constructeur initialise l'objet [Kinect](#) ainsi que les paramètres et interfaces du noeud.

```
~NodeKinectPub()
```

Private Functions

```
void init_interfaces()
```

Initialiser les interfaces du noeud.

Cette fonction initialise les interfaces du noeud en créant deux publisher pour les images RGB et infrarouge ainsi qu'un timer associé.

```
void init_parameters()
```

Initialiser la [Kinect](#) et les paramètres du noeud.

Asservissement en cap du robot SATURNE

Formule de l'erreur pour l'asservissement en cap:

$$e_{\psi} = \text{atan2}(\sin(\psi_d - \psi), \cos(\psi_d - \psi))$$

Correcteur proportionnel :

$$u = K_p \cdot e_{\psi}$$



Annexe

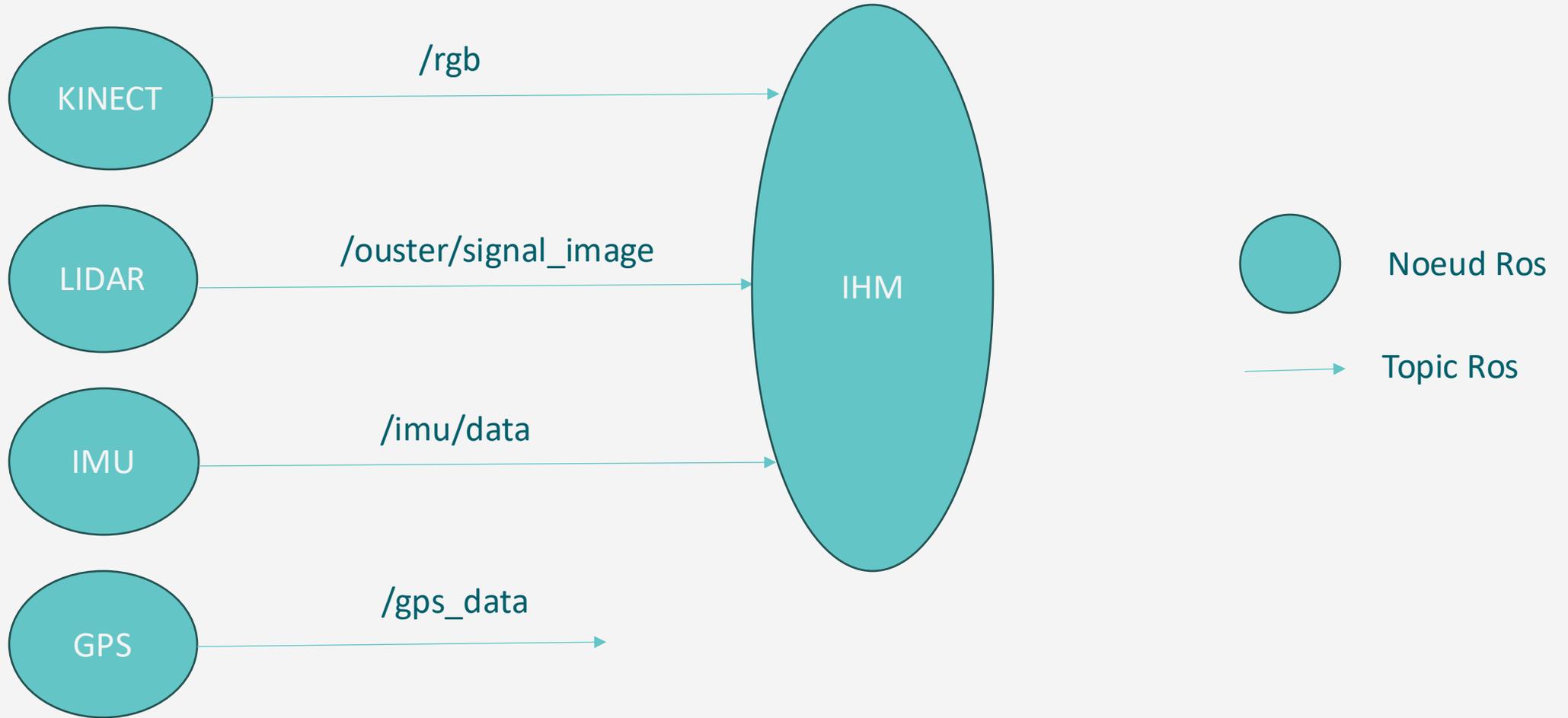


Schéma des noeuds et topic ros2