

Localisation indoor d'une fusée expérimentale

Projet de Semestre 8 – Sujet FUSEX-1



Etudiants : Alexandre Arrivé & Gabriel Fkyerat

Encadrants : Patrick Lanusse (ENSEIRB-MATMECA) & Jean-Luc Charles (ENSAM)

31/05/2022

Sommaire

CONTEXTE DU PROJET	2
OBJECTIF DE L'ETUDE.....	2
MATERIEL NECESSAIRE	3
METHODOLOGIE DE REALISATION.....	3
PRISE EN MAIN DU MATERIEL A DISPOSITION	4
COMPILATION DU LOGICIEL UWBSUPERVISOR	5
ANALYSE DES RESULTATS OBTENUS GRACE AU SOFTWARE	7
PORTAGE VERS LA RASPBERRY	8
UTILISATION DU USER GUIDE DE IIDRE	9
CARACTERISATION DU SYSTEME	10
<i>Expériences en statique.....</i>	<i>10</i>
<i>Expériences en dynamique</i>	<i>12</i>
CONCLUSION	14

Contexte du projet

Le démonstrateur Mini-APTERROS (*Advanced Propulsion Technology for Reusable Rocket and Operating System*) s'inscrit dans le cadre du projet PERSEUS (*Projet Étudiant de Recherche Spatiale Européen Universitaire et Scientifique*) de la Direction des Lanceurs du CNES (*Centre National d'Études Spatiales*). Le but de cette initiative vise est de favoriser l'émergence de solutions techniques innovantes pour des projets spatiaux. Ces projets peuvent être de nature industrielle et pédagogique.

L'ENSAM (*Ecole Nationale Supérieure des Arts et Métiers*) de Bordeaux dirige la réalisation d'un tel projet, à savoir les programmes APTERROS et Mini-APTERROS. L'objectif du programme APTERROS est de concevoir, construire et tester un démonstrateur de fusée à atterrissage vertical. APTERROS utilisera une propulsion à réaction bi liquide et sera complexe à développer.

Le programme Mini-APTERROS a pour but de développer les méthodes logicielles et les équipements électroniques nécessaire aux vols verticaux qu'APTERROS devra réaliser. Mini-APTERROS est un modèle réduit d'APTERROS qui utilise une propulsion électrique sous forme de 4 turbines. Sa structure est hautement modulaire et permet l'installation et la modification rapide de n'importe quel composant.

L'un des objectifs de Mini-APTERROS est d'analyser la trajectoire du démonstrateur en vol. Cette tâche demande de connaître la position de la fusée dans l'espace avec la plus grande précision possible. L'équipe de l'ENSAM de Bordeaux a installé un capteur Lidar (*Light Detection And Ranging*) pour connaître l'altitude précise du véhicule au niveau du sol. Notre travail se porte donc sur l'obtention de données de position horizontale.

Le véhicule est équipé de deux cartes électroniques principales. Une carte de développement Arduino Mega est utilisée afin de rassembler les données des multiples capteurs de Mini-Apterros et piloter les 4 contrôleurs des turbines ainsi que les actionneurs. Cette Arduino communique avec un micro-ordinateur Raspberry Pi 4 qui est chargée du traitement des données et du contrôle de trajectoire. La Raspberry utilise un réseau de neurones en lieu d'un système automatique classique dans le cadre des solutions innovantes avancées par le projet PERSEUS.

Plusieurs technologies de positionnement existent déjà sur le marché : GPS, Lidar, Radar, capteurs ultrasonores, etc. Cependant, il a été décidé d'utiliser un système de localisation par ondes radios du fabricant IIDRE. Ce système a été utilisé par Yoan Mollard de EirLab (fablab de l'ENSEIRB-MATMECA). Ce dernier a aussi développé du code pour de l'acquisition temps-réel et un logiciel de configuration du matériel IIDRE. A terme, le système IIDRE sera relié à la Raspberry Pi 4 afin de fusionner les données obtenues du système IIDRE avec les autres capteurs.

Objectif de l'étude

L'objectif de notre étude est de connaître la position horizontale de mini-APTERROS avec la meilleure précision possible avec le matériel de IIDRE. Cela passe par une compréhension poussée du système IIDRE, une configuration adaptée à l'environnement de test, une bonne connaissance de l'emplacement des ancres UWB et la prise en compte des contraintes de ligne de visée.

Afin d'évaluer la performance du système IIDRE, qui est conçu pour une précision maximale en extérieur et non en intérieur, une caractérisation du système est donc nécessaire selon différentes configurations, différentes positions et trajectoires de la balise, et avec différents obstacles.

Matériel nécessaire

Comme indiqué précédemment, EirLab nous a fourni le système IIDRE. Il se présente sous la forme de quatre ancrs fixées à des bras ajustables et d'une balise mobile à poser sur le véhicule. Les quatre ancrs sont alimentées par une fiche micro-USB 5V et consomment 1A. Des chargeurs fournis par EirLab peuvent être utilisés, mais l'ENSAM nous a fourni des alimentations stabilisées à 1A.

Le système IIDRE utilise les ondes radios UWB (Ultra Wide Band) pour estimer la distance entre une balise et 4 ancrs fixes. En connaissant la position de chaque balise, la balise est capable de mesurer sa distance à chacune d'entre elle, puis d'interpréter le résultat par trilatération.

La balise (ou Mobile) a l'apparence d'une clé USB et peut être branchée sur un port USB 2.0. La liaison Serial/UART établie avec l'ordinateur permet alors le transfert de données et commandes. Des incidents détaillés plus bas suggèrent que l'USB 3.0 peuvent être contre-indiqués, mais cette question n'a jamais été éclaircie. La balise a été principalement utilisée sur un ordinateur portable, mais la Raspberry Pi est équipée de ports USB femelles compatibles.

Afin de configurer le système IIDRE, Yoan Mollard nous a donné le lien d'un logiciel open-source développé par ses soins. Nommé `uwbSoftware`, il permet de configurer mais également de visualiser la position des éléments du système IIDRE dans un espace 2D. Sa compilation et son exécution, ainsi que la configuration du système, seront détaillés plus bas.

Méthodologie de réalisation

Dans un premier temps, nous allons prendre en main le matériel mis à disposition. La configuration du système IIDRE étant clé pour la précision finale de nos données, comprendre l'influence des paramètres de configuration est important. La prise en main de `uwbSupervisor` va également être abordée.

En second temps, nous allons réaliser les préparations nécessaires à la caractérisation du système IIDRE. Cela implique notamment de rédiger un programme d'acquisition spécifique afin de réaliser des analyses post-test. En utilisant `uwbSupervisor` comme un outil visuel, nous pouvons également établir des paramètres de test permettant de réduire les sources d'erreur importantes.

Dans un troisième temps, nous allons caractériser le système IIDRE. La caractérisation aura pour but d'évaluer la précision usuelle du système ainsi que les phénomènes pouvant interférer avec la mesure (obstacles sur la ligne de visée, interférences électromagnétiques, etc.). Nous allons évaluer la précision en sortie du système dans des situations où la balise est immobile, puis en mouvement.

En dernier temps, nous allons travailler vers l'application de notre code d'acquisition sur la Raspberry Pi 4 de Mini-APTERROS. Une sélection des données exploitables par le micro-ordinateur est nécessaire afin d'avoir une fusion de données optimale. Les plus brutes les données seront, le meilleur. Nous pourrons alors développer des filtres adaptés afin de lisser les anomalies de position.

Prise en main du matériel à disposition

Le matériel IIDRE peut couvrir une surface importante, avec deux pouvant être séparée de jusqu'à 100 mètres. Le système peut calculer la position de la balise dans l'espace sous 3 dimensions, mais placer les ancrs à la même hauteur permet d'avoir une meilleure précision sur les deux axes horizontaux. Cette configuration 2D est préférée dans notre cas, et sera utilisée pour tous nos tests.

Pour notre projet, les ancrs sont relativement proches, installées sur les 4 pylônes de la structure de test de Mini-APTERROS. Puisque l'antenne de chaque balise et ancre transmet à une puissance élevée (33 dBi par défaut), un bruit thermique important peut apparaître. Nous avons donc configuré la balise et les ancrs pour émettre à une puissance plus faible (10 à 15 dBi).

Avec l'aval de Michel Seyrac, ingénieur et créateur du système IIDRE, nous utilisons le canal de transmission n°5. Selon lui, la fréquence centrale et la largeur de bande passante devraient permettre la meilleure précision dans un environnement intérieur, fortement pollué par les ondes électromagnétiques.

Toujours selon les recommandations de M. Seyrac, les quatre ancrs ont été placées dans le même plan. Comme précisé précédemment, cette configuration 2D permet une meilleure précision sur les axes horizontaux. Nous avons placé les ancrs de sorte à ce qu'elles soient au plus proche de l'altitude maximale prévue de Mini-APTERROS. Ainsi, le véhicule reste dans la proximité des ancrs, mais la ligne de visée entre la balise du démonstrateur n'est pas obstruée par le châssis.

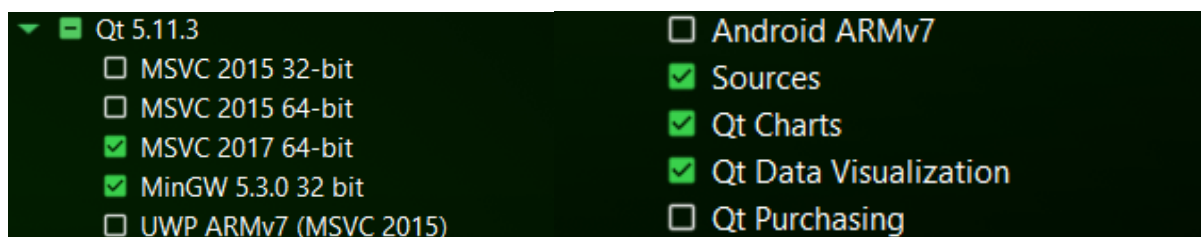
Afin de calculer sa position dans l'espace, la balise embarquée sur Mini-APTERROS doit connaître la position de chaque ancre fixe. Avec un mètre, nous avons mesuré la distance entre ces ancrs, puis avons dressé un repère pour les localiser dans un plan horizontal. Ces positions peuvent alors être entrées dans un fichier .json du logiciel uwbSupervisor, ou manuellement via commandes AT.



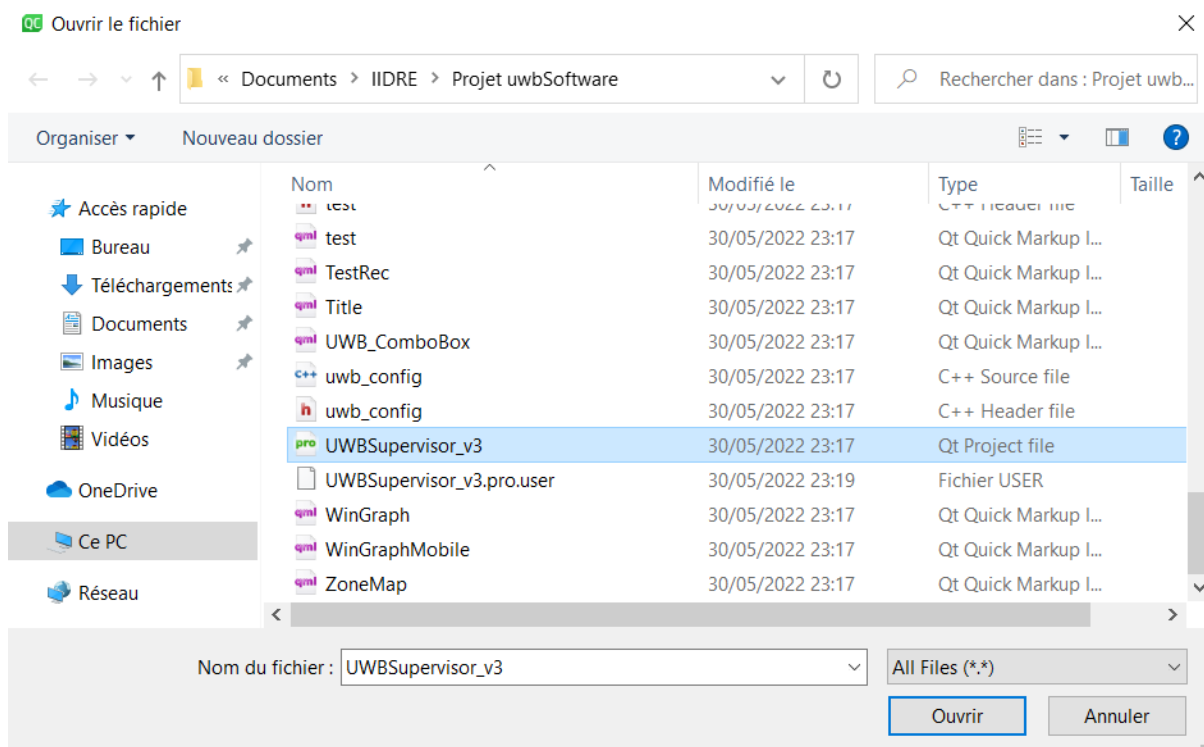
Compilation du logiciel uwbSupervisor

Avant de pouvoir exploiter les données de position, nous pouvons les visualiser en utilisant le logiciel uwbSupervisor. Il permet de facilement configurer le système IIDRE, représenter la position de chaque ancre et balise active, et tracer la position de la balise en temps réel. Cependant, le logiciel est fourni comme code source, et un fichier précompilé n'est pas inclus. Nous devons donc le compiler de manière appropriée afin de l'utiliser, en utilisant le programme QtCreator.

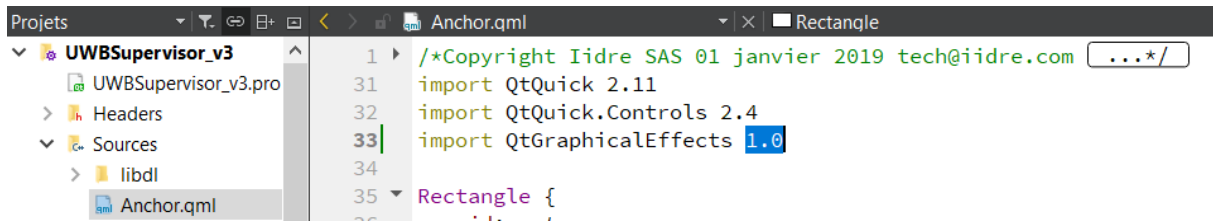
QtCreator est un outil de création d'application, permettant de joindre du code à une interface graphique et d'autres outils de visualisation. Dans le cas de uwbSupervisor, l'application a été conçue sous la version 5.11 de Qt, compilée avec la suite MinGW (version 5.11) et les packages optionnels Qt Charts, Qt Data Visualisation et Sources. Ces packages peuvent être ajoutés via le Qt Maintenance Tool, rubrique Ajout de Composants -> Archives.



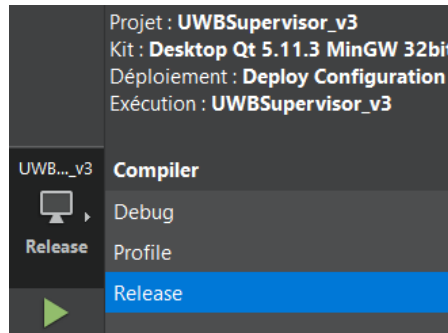
Le code source de uwbSupervisor a été obtenu sur le repository GitHub de IIDRE, et a été cloné sur une machine Linux (une machine Windows peut également être utilisée). Le code source peut être téléchargé ici : <https://github.com/IIDRE/uwbSupervisor>. Afin de compiler et lancer uwbSupervisor, il faut modifier un fichier sur QtCreator. Tout d'abord, il faut ouvrir le fichier .proj associé à uwbSupervisor.



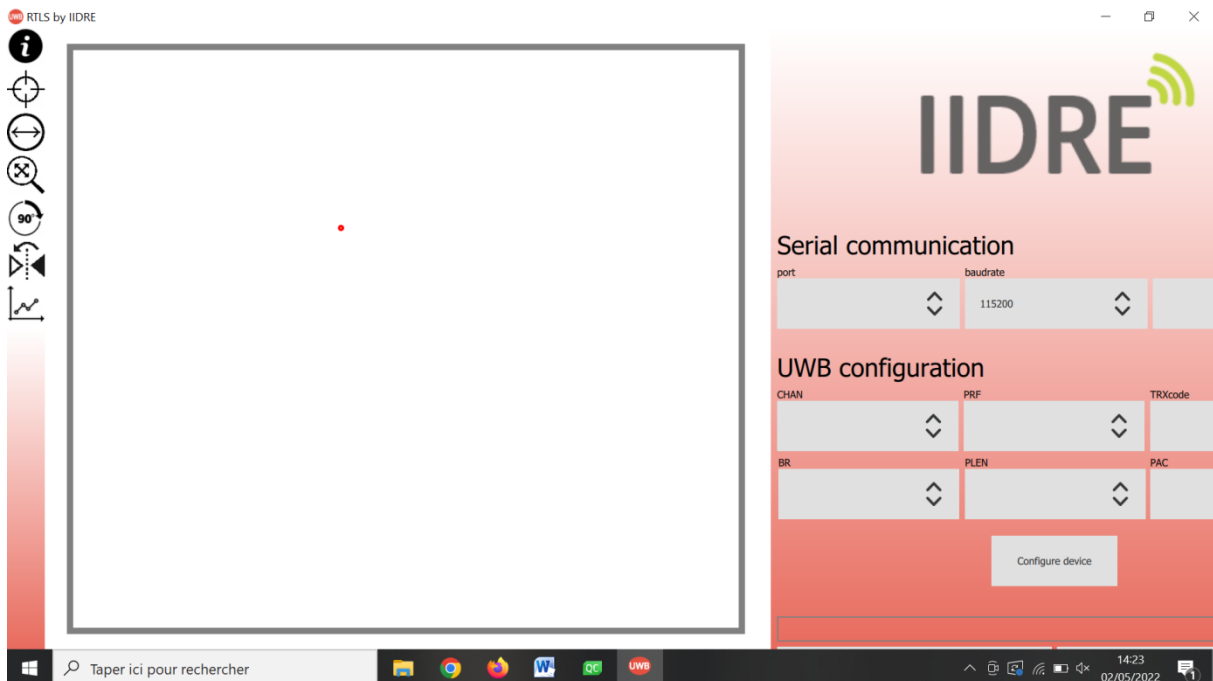
Dans l'arborescence du projet, sous les Sources, le fichier "Anchor.cpp" doit être modifié afin d'utiliser une version antérieure du package QtGraphicalEffects, passant de 1.12 à 1.0.



Enfin, le projet doit être compilé puis lancé sous sa configuration Release.



Selon la machine et le système d'exploitation, l'interface visuelle de uwBSupervisor peut être incomplète. Sur trois ordinateurs testés, une machine a présenté une interface zoomée incomplète comme celle figurant ci-dessous :



Cependant, dans la majorité des cas, l'interface présente l'ensemble des boutons permettant la configuration du système. Les instructions sont disponibles sur le GitHub de Yoan Mollard. Grâce à ce software, nous pouvons modifier la configuration du système IIDRE. Changer de canal de communication se fait en modifiant la case « CHAN ». Dans nos expériences, nous avons seulement modifié le canal de communication ainsi que le débit de communication en modifiant « BR » (bit rate).

Analyse des résultats obtenus grâce au software

Le logiciel uwbSoftware nous a servi à évaluer les performances du système IIDRE dans différentes configurations et de tester la qualité des relevés obtenus. L'atout principal du logiciel est de pouvoir visualiser les relevés temporels de manière instantanée. De plus, les points s'effacent au fur et à mesure ce qui permet de voir si les relevés de la position de la balise se stabilisent.

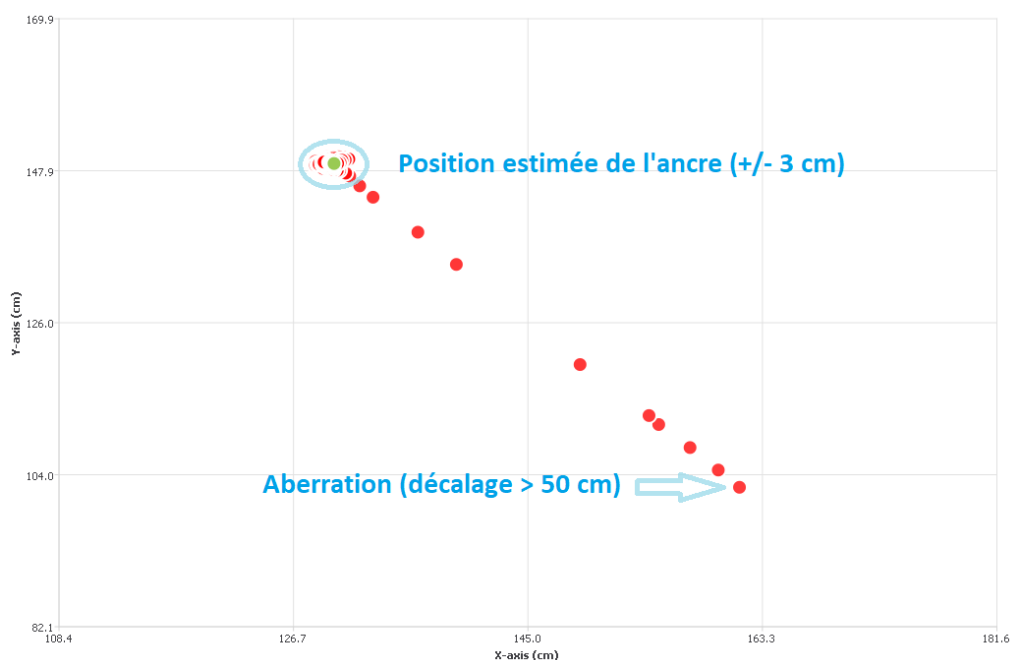
Le premier test effectué a été fait en salle de TP automatique à l'ENSEIRB-MATMECA. Le système était positionné dans un environnement doté de beaucoup d'obstacles et soumis à de fortes perturbations électromagnétiques. Des ordinateurs avec des écrans venaient faire obstacle entre les ancres et la balise. De nombreux équipements haute-puissance étaient également en fonctionnement.

Les résultats étaient peu encourageants : alors que la balise était immobile, sa position pouvait réaliser des sauts de plus de 20 cm. La balise était immobile mais sa position estimée n'arrivait pas à se stabiliser autour d'un point d'équilibre. La première démarche que nous avons entreprise est d'obtenir un niveau de cohérence élevé entre la position réelle de la balise et les relevés exposés sur uwbSupervisor.

Pour cela, nous nous sommes placés dans l'environnement de test attendu, au centre de la structure où est situé Mini-APTERROS. La balise était toujours maintenue immobile. Là aussi les relevés temporels présentaient une erreur importante d'environ une vingtaine de centimètre. Nous nous sommes rendu compte que cette erreur résultait de l'interaction entre la balise et le PC utilisant uwbSupervisor.

La balise se connecte sur un port USB de l'ordinateur portable, elle est donc très proche de sa structure métallique. La balise avait donc une ligne de visée constamment obstruée vers au moins une des ancres du système, parfois plus. De même, le chargeur de l'ordinateur portable était branché, et débrancher l'ordinateur a également amélioré la stabilité de la position estimée de la balise.

La solution a été d'utiliser une rallonge USB, fournit par M. Charles, qui permet d'éloigner l'ordinateur de la balise et ainsi d'éloigner un obstacle majeur et une source d'ondes électromagnétiques. Nous avons alors pu réaliser une acquisition de données avec ces correctifs. Le but de cette acquisition est de pouvoir observer la stabilité et la précision du système dans le cas optimal, en statique sans obstacles.



Comme on peut le voir sur le graphique obtenu sur uwbSupervisor, la majorité des relevés de position sont concentrés dans un cercle de 3 cm de rayon. Cependant, on constate également que certains relevés sont à 50cm de la position estimée. La précision actuelle n'est pas mauvaise, mais l'objectif est d'arriver à une précision de l'ordre du centimètre. Ce test a été effectué en utilisant le canal de communication n°2 alors qu'il est plus adéquat d'utiliser le canal n°5 pour augmenter la précision.

L'aberration que nous avons obtenue nous laissait perplexe. Aucun objet ne venait faire obstruction entre l'une des ancres et la balise, et tout objet pouvant émettre des ondes électromagnétiques importantes ont été éloigné du dispositif. Il advient que l'erreur venait du matériel. Comme on peut le voir sur le graphique ci-dessus, l'aberration suit une trajectoire diagonale allant vers une direction bas-droite.

Cela laisse supposer que l'une des ancres cesse momentanément de communiquer, le temps d'un seul relevé de distance par la balise. L'ancre exacte a été identifiée, et sa position dans le repère établi plus tôt correspond à la direction de l'aberration. Le chargeur utilisé pour alimenter la balise n'était pas en bon état, nous sommes alors passés sur les alimentations 5V 1A de l'ENSAM. Cette erreur a cessé d'apparaître.

Portage vers la Raspberry

L'utilisation du logiciel uwbSupervisor se limite à l'analyse rapide des relevés de position de la balise et l'élimination des sources d'erreurs importante. La caractérisation et l'exploitation du système IIDRE, cependant, nécessite le portage de code Python sur un ordinateur puis la Raspberry Pi 4. Pour cela, nous allons écrire du code Python s'appuyant sur un exemple de code temps-réel développé par Yoan Mollard.

Pour pouvoir réaliser le portage, il faut d'abord comprendre comment la balise renvoie sa position à l'ordinateur. Jusqu'ici, le logiciel uwbSupervisor a masqué la manière dont la balise communique, et son interface graphique ne permet pas de modifier un certain nombre de paramètres. Le matériel IIDRE est aussi capable de renvoyer des données sous différents formats, certains étant plus ou moins filtrés. Il faut que l'on analyse ces différentes traces afin de déterminer la plus adéquate aux besoins du projet.

Nous allons maintenant utiliser les commandes AT du système IIDRE pour venir configurer la balise seule. Dans un premier temps, il faut analyser le manuel utilisateur livré par IIDRE. Ce manuel contient l'ensemble des commandes permettant de configurer la balise, modifier les traces renvoyées vers l'ordinateur et réinitialiser le système.

Après cela, nous allons développer un script Python permettant de réaliser une acquisition de données envoyées par la balise. Des corrections et suppressions d'erreur pourront être développées afin de les exploiter le plus possible. L'idée finale étant de pouvoir faire la fusion de données entre celles récupérées avec la balise et celles du LiDAR pour qu'il y ait un post-traitement par la suite.

Utilisation du User Guide de IIDRE

Le logiciel uwbSoftware permettait de faire des configurations basiques de la balise mais surtout de pouvoir placer la balise et donc la position du mini-Apterros dans un plan. Comme expliqué précédemment, il ne permet pas d'examiner ou de changer la manière dont la balise envoie ses données à l'ordinateur. Chaque trame envoyée par la balise de manière non-sollicitée est une trace.

L'utilisation de guide utilisateur permet de découvrir les différentes traces utilisables. En effet, IIDRE offre plus d'une dizaine de traces, permettant d'accéder à plusieurs fonctionnalités du système IIDRE. Les 3 traces principales dans le cadre du projet sont DIST, DIST_DBG et MPOS. Chacune présente un certain format, mais elles s'organisent d'une manière commune : timestamp (en ms depuis l'allumage de la balise), numéro d'identification de la balise, et les données spécifiques à la balise et la trace.

La trace DIST renvoie la distance mesurée à l'instant du timestamp entre la balise et une des 4 ancrs connectées. Ces données sont accompagnées de la position estimée de la balise sous des coordonnées X, Y, Z ainsi que de la puissance de l'onde reçue et de deux indicateurs de qualité. La position estimée a été filtrée par la balise en utilisant un algorithme de trilatération et de la fusion de données interne (usage d'une centrale BNO 055), et la distance mesurée est passée à travers un filtre.

1. +DIST

Message with the exchanges synthesis (filtered data) between a tag and one anchor. These distances are the ones used by the multilateration algorithm.

```
+DIST:<TMSTP>,<ANCHOR_UID>,<DIST>,<X>,<Y>,<Z>,<FP_PWR_LVL>,<IDIFF>,<MC>
```

La trace DIST_DBG est identique en format, mais la distance mesurée n'est pas filtrée. Dans le cas où l'une des ancrs ne répond pas à une mesure de distance, le timestamp prendra la valeur 999999.

2. +DIST_DBG

Message with the exchanges synthesis (raw data) between the tag and one anchor. If an anchor is not responding, the TMSTP field will be 999999, indicating a time-out.

```
+DIST_DBG:<TMSTP>,<ANCHOR_UID>,<DIST>,<X>,<Y>,<Z>,<FP_PWR_LVL>,<IDIFF>,<MC>
```

La trace MPOS permet seulement d'obtenir la position estimée en coordonnées X, Y et Z ainsi que les vitesses VX, VY et VZ estimées. Dans notre cas, puisque nous opérons sur une configuration 2D, les vitesses indiquées sont nulles.

3. +MPOS

Message with the tag position.

```
+MPOS:<TMSTP>  
,<X_COORD>,<Y_COORD>,<Z_COORD>  
,<X_VELOCITY>,<Y_VELOCITY>,<Z_VELOCITY>
```

Une distinction importante : puisque nous utilisons une configuration en 2D et que toutes les ancrs sont à la même hauteur, le système a du mal à interpréter la coordonnée Z. Ainsi, toutes les trames renvoyant la coordonnée Z observeront un caractère nul (\0) à la place d'une valeur. Cela peut interférer avec des programmes de « parsing » en Python.

L'usage de ces différentes traces se fait en configurant la balise via les commandes AT. Les commandes AT sont utilisées pour communiquer avec du matériel sur une liaison Serial. Afin d'utiliser les commandes AT, il faut ouvrir une liaison Serial. Nous avons fait cela avec le logiciel Putty, et la liaison s'est faite avec un Baud Rate de 9600. Le port choisi doit être celui sur lequel la balise (ou l'ancre) est branchée, et le nom dépend de votre système d'exploitation. Sous Linux, le port utilisé était généralement /dev/ttyUSB0 ou /dev/ACM0. Cette liaison doit être fermée après chaque configuration.

D'autres commandes servant à la configuration du matériel IIDRE sont indiquées. Tout d'abord, on peut rentrer manuellement les coordonnées de chaque ancre en utilisant la commande *AT+POS*. Cela permet de configurer notre repère virtuel sans avoir à passer par *uwbSupervisor*. Dans le cas d'un reset d'usine (à éviter, *AT+RESET=3*), la commande *AT+NODE* peut être utilisée pour ajouter des ancres et des balises au réseau utilisé par le système.

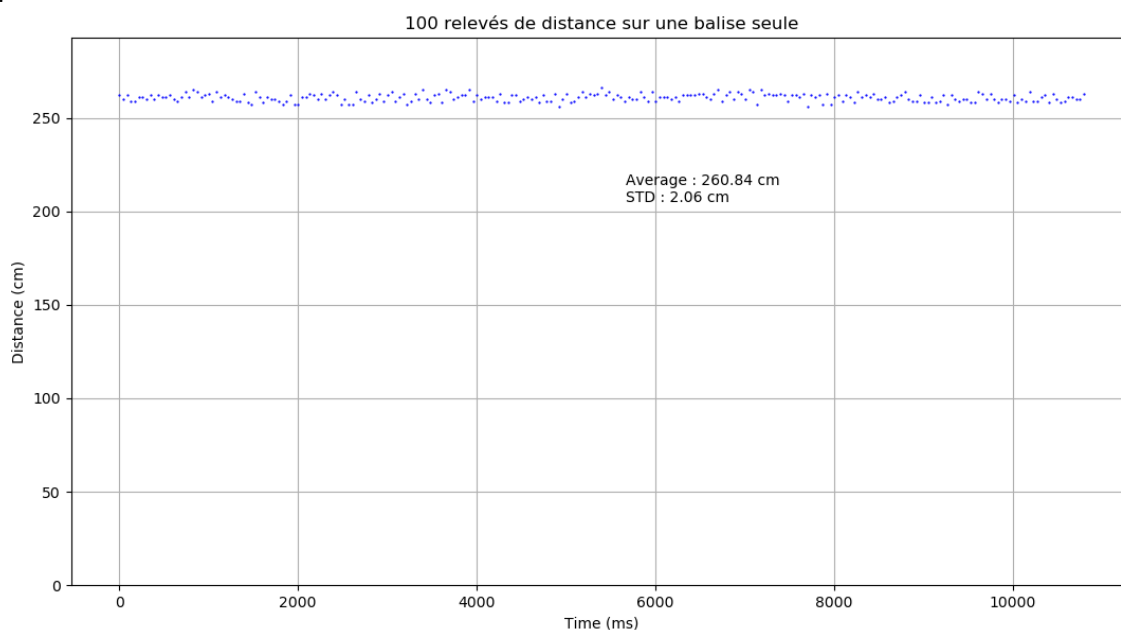
Nous avons aussi utilisé la commande *AT+CHAN* qui permet de changer de canal de communication. Néanmoins, il faut que toutes les ancres et la balise communiquent sur le même canal, il faut donc venir configurer individuellement chaque ancre et la balise avec des commandes AT afin de changer de canal sans passer par *uwbSoftware*. Cette erreur particulière a contribué à un problème matériel qui nous a bloqué pendant quelques semaines.

Caractérisation du système

Expériences en statique

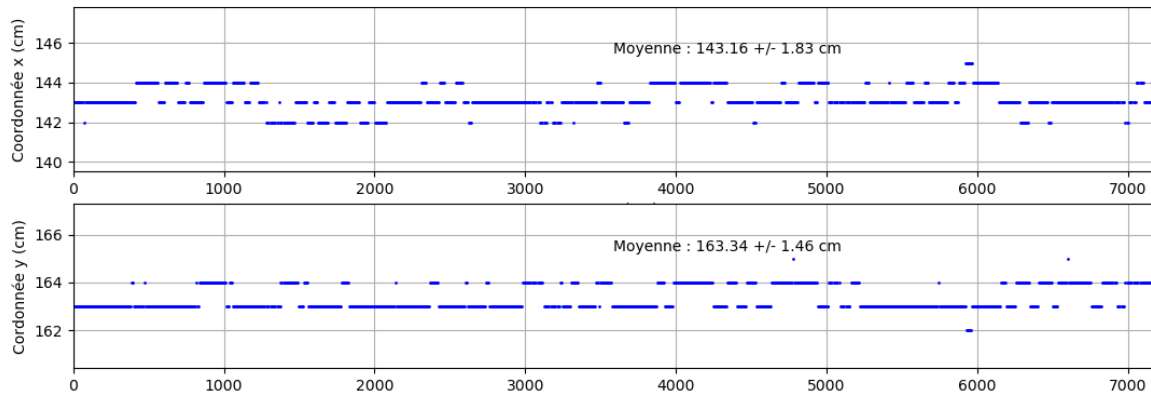
Après avoir configuré la balise et les ancres, et en connaissant les trames de sorties, nous pouvons exploiter les données observées. Dans un premier temps, nous n'allons utiliser qu'une ancre qui va communiquer avec la balise, et nous allons effectuer plusieurs mesures de distances pour estimer la précision de la distance mesurée. Cette expérience n'a pas été faite en utilisant le canal 5, mais le canal 2.

Nous avons développé un programme Python se servant de celui de monsieur Mollard permettant de récupérer les données de distance en utilisant la trame DIST.



On peut voir que ce test montre une précision proche de notre objectif. Les valeurs en sortie mesurées en centimètres entiers. Ici, pour 100 relevés de distances, nous avons un écart-type de 2.06 cm pour une communication avec le canal 2. Il faut aussi prendre en compte que cette mesure est faite sur une balise envoyant des trames filtrées DIST.

Voyons voir les résultats obtenus en utilisant le canal de communication n°5 et en connectant les 4 ancrs. Nous allons utiliser la trame MPOS pour observer la position horizontale. La première manipulation se fait avec la balise qui est statique. L'axe horizontal est le temps en millisecondes.

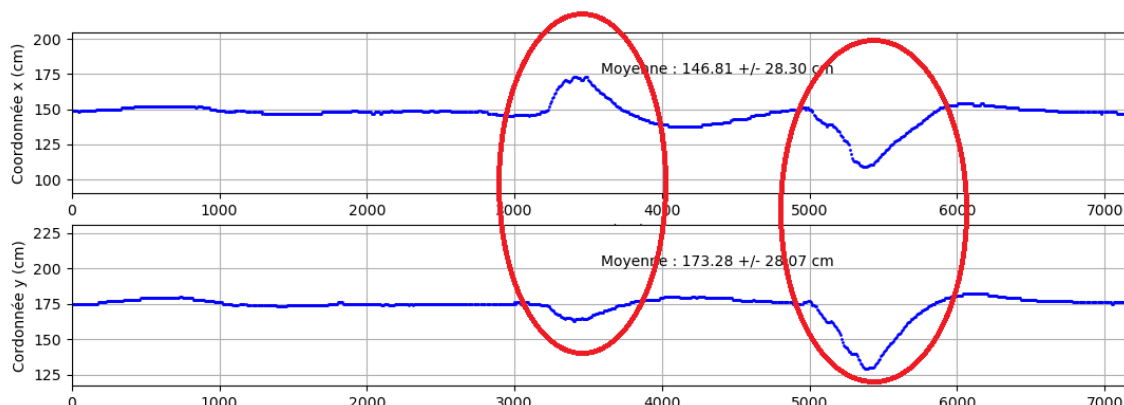


La trame MPOS permet de positionner la balise dans l'espace en 3D. Dans le cas du projet, le plan horizontal XY est notre priorité. L'erreur indiquée sur le graphe est mesurée à 3σ (3 fois l'écart-type), indiquant que chaque mesure individuelle à une probabilité de 99,9% de se situer dans le domaine d'erreur indiqué. Cette méthode d'estimation est bien plus rigoureuse que nos observations sur uwbSupervisor.

Concrètement, on observe que la précision est meilleure : selon les axes X et Y, l'erreur à 3σ est de moins de 2cm. Comme on peut le voir, les coordonnées étant des entiers introduisent une granularité à la position qui diminue la précision absolue, mais la position moyennée sur 1000 valeurs est bonne en statique. A noter que ces données sont obtenues par fusion de données interne avec la centrale BNO 055 de la balise, et ne sont donc pas optimales dans notre objectif d'obtention de données brutes.

Effectuons maintenant un deuxième test où la balise est toujours immobile mais un humain passe entre les ancrs et la balise, entraînant une perte de vision avec deux ancrs successives.

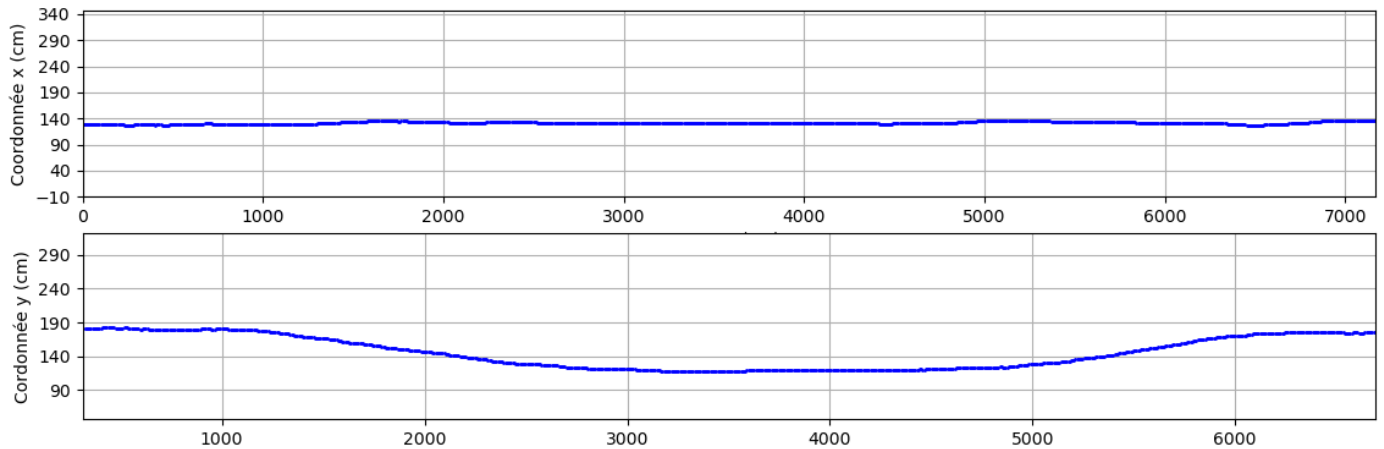
passage d'un individu entre une balise et l'ancre



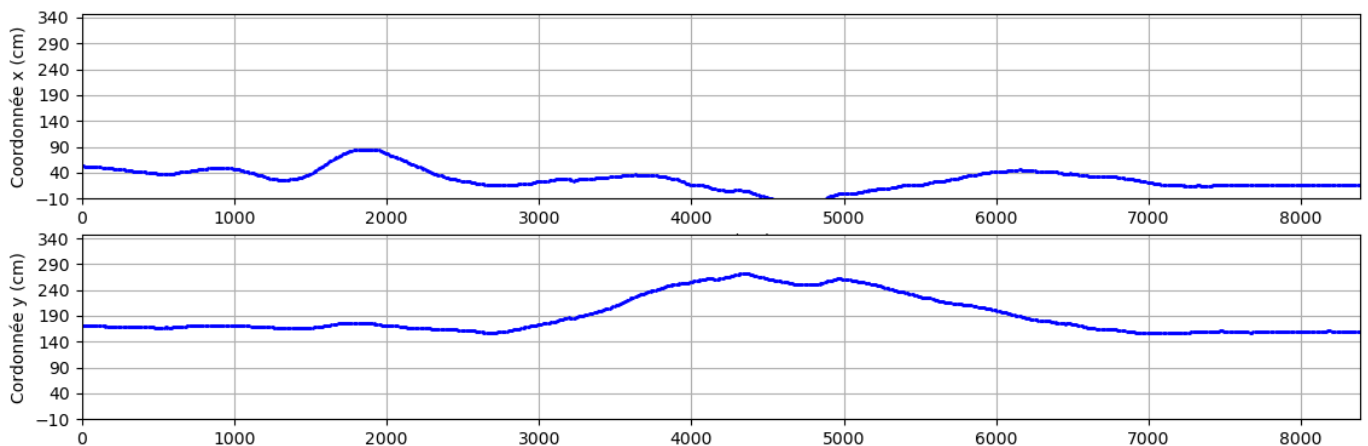
Lors de ce deuxième test, on constate à quel point le système est sensible à une obstruction de ligne de visée. Lorsqu'un obstacle traverse la ligne de visée, une perturbation allant jusqu'à 45cm sur l'axe Y se produit, alors que la balise est immobile. Ces observations réaffirment nos craintes quant à la position de la balise sur le démonstrateur : il faut maintenir la ligne de visée à tout prix si l'on souhaite avoir un relevé de position précis.

Expériences en dynamique

Maintenant, effectuons des tests où la balise est en mouvement. Nous avons commencé par des translations rectilignes selon l'axe Y, puis avons utilisé des trajectoires semi-circulaires.



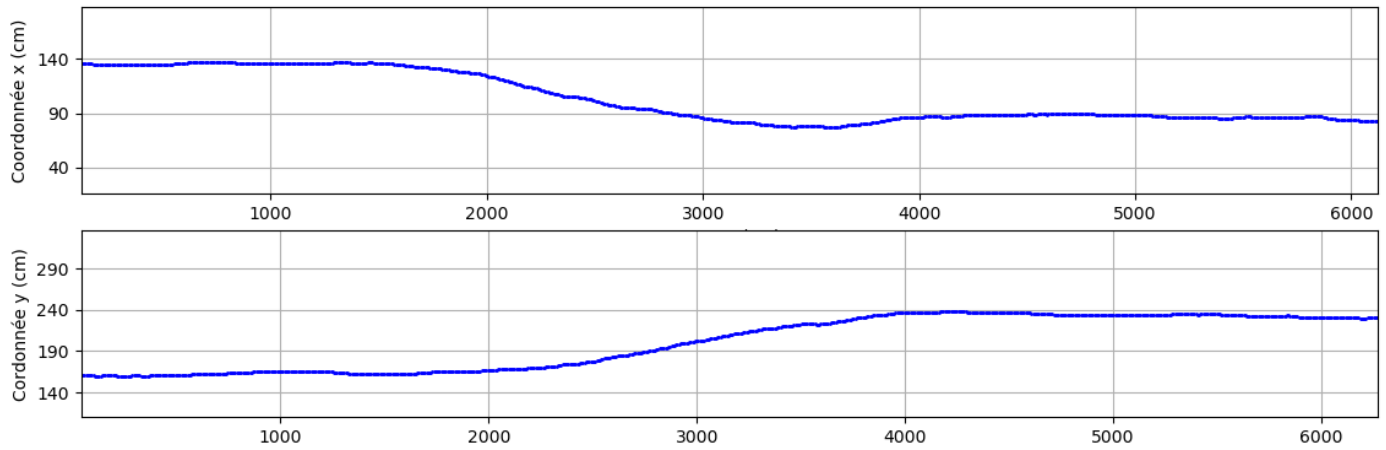
La trajectoire de la balise dans l'exemple ci-dessus est une translation aller-retour de 42 cm dans la direction Y- de notre repère. Cette distance est mesurée expérimentalement à l'aide d'un mètre à ruban tenu à la main durant la translation. Cependant, la translation observée par le système IIDRE est mesurée à 59 cm, soit une erreur ponctuelle de 17 cm. La mesure est consistante, puisque la position est stabilisée lorsque la balise est presque statique pendant les phases de repos.



Une seconde mesure sur le même axe, mais avec une transition plus rapide et plus ample (90 cm), révèle un autre comportement étrange. Durant la manipulation, en phase de repos (avant 2500 ms), la position sur l'axe X varie fortement alors que la balise est encore statique. Bien qu'elle soit tenue à la main, les écarts dus au tremblement de l'opérateur ne peuvent être supérieurs à 10 cm, c'est pourquoi des écarts de 45 cm sont étranges.

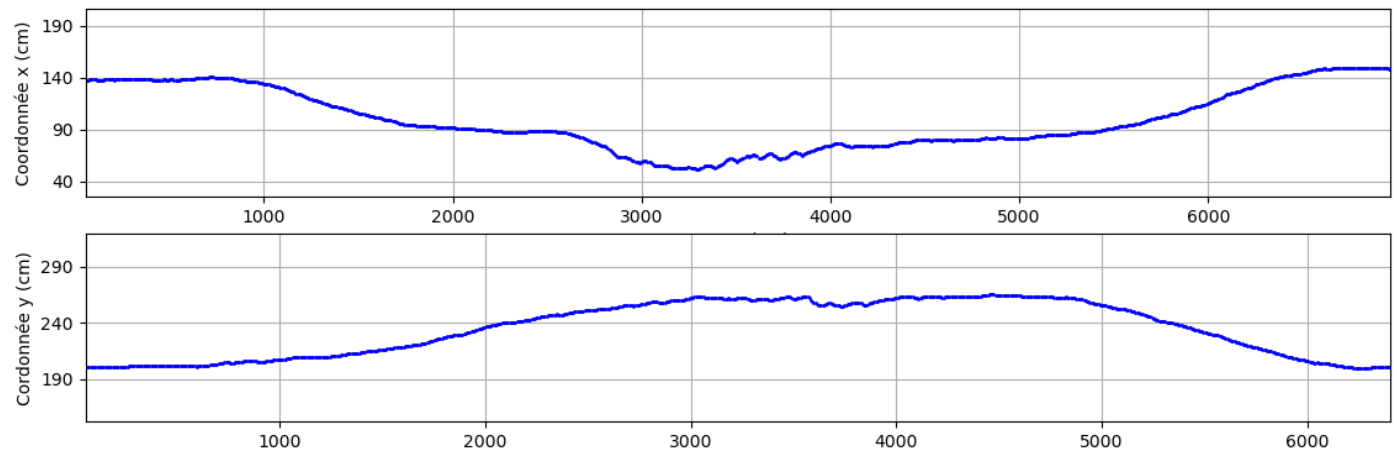
Par ailleurs, l'erreur sur l'axe Y est estimée à 15 cm sur la translation. Nous avons réalisé près d'une dizaine de translations sur les axes X et Y, en orientant la balise sous différentes orientations, mais l'erreur relative mesurée ne descend jamais en dessous de 10 cm.

Vers la fin de notre étude, nous avons tenté une autre manipulation dans l'espace. Nous avons réalisé une rotation à rayon constant autour de l'axe Z. L'objectif était de voir si le type de mouvement influençait la précision de la mesure de position. La rotation s'est effectuée à bout de bras, dans le sens des aiguilles d'une montre, à une distance de 50 cm du centre de pivot. La rotation était d'environ 90 degrés.



La variation de position sur l'axe X est très consistante avec le mouvement, on retrouve bien un déplacement de 50 cm après un léger rebond. Cependant, la mesure sur l'axe Y suggère un déplacement de 80 cm, soit une erreur de 30 cm.

Une seconde rotation a été effectuée à une vitesse plus élevée.



L'erreur observée sur l'axe Y est plus faible, estimée à 15 cm, mais la mesure sur l'axe X présente maintenant un rebond fortement perturbé. Avant ce rebond, la mesure sur l'axe X s'était correctement stabilisée à 50 cm de sa position initiale. L'origine de ce rebond, ou du moins la perturbation, est inconnue.

Dans tous les cas étudiés, les mesures statiques de position en utilisant la trace MPOS sont d'une précision insuffisante à la localisation centimétrique. Les tests préliminaires réalisés sur la trace DIST suggèrent que ce critère ne sera pas non plus respecté avec une simple trilatération, mais des essais plus poussés doivent être entrepris. Quant au comportement dynamique, la trace MPOS présente des erreurs inconsistantes et importantes que nous ne sommes pas parvenus à restreindre avant la fin de notre étude.

Conclusion

Notre étude du système IIDRE nous a permis d'appréhender son fonctionnement et ses différentes configurations. Bien que nous ayons tenté de réduire les sources de bruit, perturbations et toute autre élément introduisant une erreur, nous n'avons pas été capable d'atteindre une précision de l'ordre du centimètre dans aucun de nos tests.

La seule possibilité restante permettant potentiellement d'atteindre cette précision serait de réaliser une trilatération à partir de la trace brute DIST_DBG et d'accompagner la mesure d'un filtre performant. La trilatération peut être accomplie avec le package Python scipy, et les mesures de distances brutes peuvent être obtenues de la balise en utilisant une modification du code Python actuel. Quant au filtre, nous n'avons pas pu avancer nos travaux à ce stade.