



---

# Projet IAR M2 ANDROIDE - Algorithme automatique de calibrage pour le mouvement de pogobots

---

*Étudiants :*

Baris KAFTANCIOGLU : 28711733

Paul-Tiberiu IORDACHE : 28706827

*Encadré par :*

Nicolas BREDECHE

21 janvier 2025

# 1 Introduction

L'objectif de notre projet est d'implémenter un algorithme de calibration automatique et de mener une étude expérimentale sur les différentes méthodes de calibration automatique permettant au robot de marcher en ligne droite. Cette étude porte sur les Pogobots (figure 1), équipés de différents types de squelettes : un exosquelette imprimé en 3D (figure 2 a, b)) et des brosses à dents (figure 2 c, d)). Une analyse de leurs performances est également réalisée.

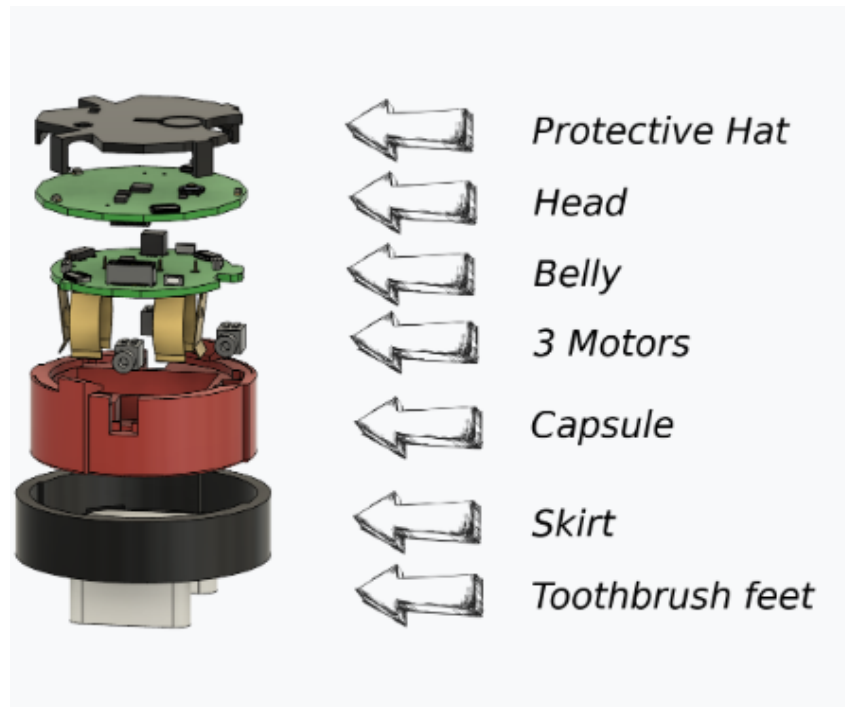


FIGURE 1 – Composants d'un pogobot

L'algorithme de calibration de base utilise un filtre de Kalman étendu (EKF) pour ajuster les moteurs d'un robot en fonction des données des capteurs. L'objectif principal est de minimiser l'écart de trajectoire afin d'assurer un mouvement rectiligne du robot. Cet algorithme se compose de trois grandes étapes :

- Initialisation du filtre de Kalman ;
- Boucle de calibration ;
- Vérification et ajustement final.

À l'étape de l'initialisation, les paramètres initiaux sont configurés en fonction de la puissance des moteurs. Les matrices d'état, de covariance et de bruit de processus sont initialisées pour modéliser l'incertitude des mesures.

Avant de nous décider sur l'approche à choisir, nous avons exploré plusieurs idées concernant l'algorithme de calibration. En testant une première approche, nous avons constaté de nombreuses sources d'incertitude liées aux Pogobots. Même après le passage par un filtre de Kalman, les données obtenues par les capteurs du robot restent très bruitées (par exemple, lorsque le robot est immobile, la valeur de la vitesse angulaire n'est pas nulle).

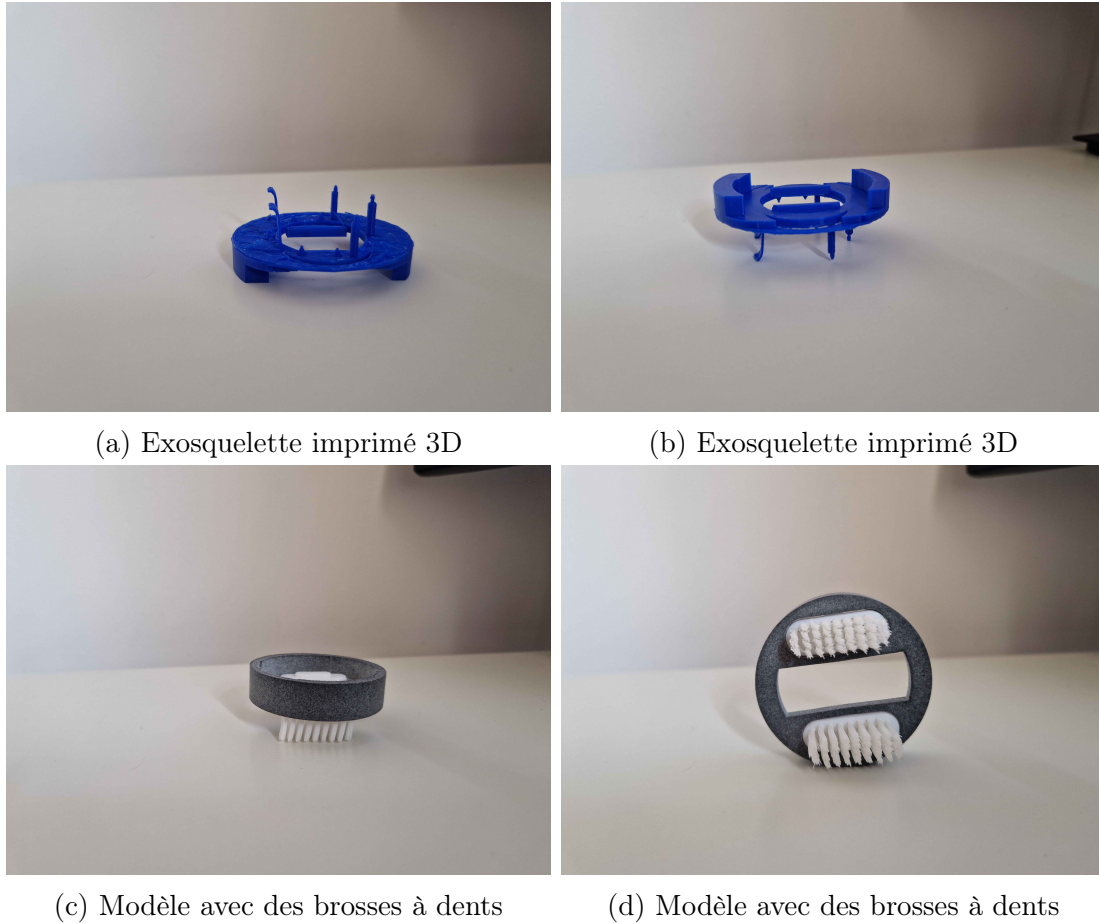


FIGURE 2 – Les différents modèles de squelettes

De plus, dans le cas de nos deux squelettes, lorsque le contact entre le Pogobot et le corps de l'exosquelette n'est pas parfait, ou si les brosses à dents ne sont pas correctement fixées, les vibrations des moteurs, qui permettent le déplacement du robot, ne sont pas transmises correctement. Cela engendre plusieurs phénomènes indésirables dus à ces incertitudes.

Ces observations nous ont conduit à tester plusieurs méthodes de calibration en exploitant les différents capteurs, afin de choisir l'approche qui, malgré ces incertitudes, permettrait de résoudre efficacement le problème de calibration de notre robot.

Notre code (dossier exemples, [pogoKalman/kalman.c](https://github.com/pogobot/kalman) et [DemoCalibration/demo.c](https://github.com/pogobot/demoCalibration)), ainsi que les vidéos et les données collectées lors de nos tests, sont disponibles sur notre [GitHub](https://github.com/pogobot).

## 2 Les différents algorithmes de calibration testés

### 2.1 Calibration par la variation de la vitesse angulaire

Les premières études expérimentales de ces approches ont été réalisées sur 15 itérations, chaque essai ayant une durée de 750 ms pour récupérer les données filtrées par le filtre de Kalman.

### 2.1.1 Calibration en augmentant les valeurs de moteurs avec une valeur constante

Dans la boucle de calibration, les moteurs gauche et droit sont activés pour une durée spécifique. Les capteurs gyroscopiques (**gyro**) sont lus pendant que le robot se déplace à l'aide de la fonction `pogobot_imu_read`. Le filtre de Kalman prédit l'état futur en utilisant les modèles de mouvement, puis met à jour cet état en fonction des observations des capteurs, réduisant ainsi l'erreur. Le biais est corrigé en comparant les données filtrées (par exemple, la valeur **gyro\_z**) aux attentes. Les valeurs de puissance des moteurs sont ensuite ajustées pour compenser les écarts de trajectoire.

Les valeurs des moteurs sont contraintes à rester dans une plage valide. Des pauses sont insérées pour stabiliser le robot avant de réévaluer sa trajectoire.

Lors de nos campagnes de test, nous avons obtenu des résultats très différents avec les mêmes paramètres. En utilisant la même approche, nous avons observé les trois types de comportements suivants :

1. **Calibration complète** : le robot se déplace en ligne droite. Cependant, ce cas est rare.
2. **Déviations vers la gauche** : dans la plupart des cas, le robot a tendance à se déplacer vers la gauche, ce qui s'explique par le fait que le moteur gauche est plus puissant que celui de droite (même si, dans le code, la valeur attribuée au moteur droit est plus élevée après la calibration).
3. **Phénomène de drift** : le robot maintient une orientation fixe, mais dérive latéralement vers la gauche ou la droite, le plus souvent vers la gauche. Cette dérive est due à la puissance supérieure du moteur gauche, même si, dans le code, la valeur attribuée au moteur droit est plus élevée.

### 2.1.2 Calibration stabilisée sur l'axe de rotation

Dans cette section, nous adoptons une approche similaire à celle de la section 2.1.1. Cependant, en raison du fort bruit présent dans nos capteurs, même après le passage par le filtre de Kalman, nous observons des valeurs différentes de zéro, même lorsque le robot reste parfaitement droit. Pour remédier à cela, nous avons introduit une variable epsilon, qui sert de seuil pour les valeurs du gyroscope.

Lorsque la valeur filtrée dépasse ce seuil, nous appliquons l'approche décrite en section 2.1.1. En revanche, si la valeur reste inférieure au seuil, un compteur enregistre le nombre de fois consécutives où cette condition est vérifiée. Si cette situation se répète un certain nombre de fois, nous conservons les valeurs actuelles des moteurs et arrêtons la calibration.

En ce qui concerne les études expérimentales, nous avons observé des comportements très similaires à ceux décrits dans la section 2.1.1. Cependant, le choix de la valeur d'epsilon reste très sensible en raison du fort bruit présent dans ces capteurs. Ainsi, lorsque la valeur est très petite, la calibration se termine prématurément, tandis qu'une valeur d'epsilon un peu plus grande revient à reproduire l'approche décrite en section 2.1.1.

Finalement, bien que cette approche produise des résultats similaires, elle reste tout de même biaisée par la valeur d'epsilon. Ainsi, même si cette valeur est inférieure à notre seuil, la calibration s'arrête, et le robot ne parvient toujours pas à se déplacer en ligne droite.

### 2.1.3 Calibration en augmentant les valeurs de moteurs avec une valeur proportionnelle à la vitesse angulaire

Dans cette section, nous proposons une approche où les puissances des moteurs sont ajustées en fonction des valeurs du gyroscope sur l'axe  $z$ . Nous calculons ces valeurs pour déterminer si le robot dévie vers la gauche, vers la droite, ou reste stable. Nous multiplions la valeur du gyroscope par une constante, puis décidons des ajustements selon les cas suivants :

- Si la valeur du gyroscope est négative, c'est-à-dire que le robot dévie vers la gauche, nous réduisons la puissance du moteur gauche et augmentons celle du moteur droit.
- Si la valeur du gyroscope est positive, c'est-à-dire que le robot dévie vers la droite, nous réduisons la puissance du moteur droit et augmentons celle du moteur gauche.

Cette approche vise à minimiser les déviations. Cependant, nous n'avons pas observé d'améliorations dans le comportement du robot par rapport à ceux décrits dans la section 2.1.1. Cela est, une fois de plus, dû au bruit présent dans les capteurs. En effet, si la valeur du gyroscope ne reflète pas la réalité, mais qu'elle est soit exagérée soit inversée (par exemple, même si le robot se déplace vers la gauche avec une valeur  $< 0$ , le gyroscope peut indiquer une valeur  $> 0$ ), les valeurs des moteurs sont alors modifiées proportionnellement à cette lecture erronée. Lorsque cela se produit, les ajustements des moteurs sont incorrects, ce qui empêche la calibration d'aboutir à un bon résultat.

En revanche, nous avons observé qu'en augmentant la durée de chaque essai pendant la calibration de 750 ms à 2 secondes, la qualité de la calibration obtenue s'améliore par rapport à celle réalisée avec 750 ms. L'augmentation du temps de chaque essai sera étudié plus en détail dans ce rapport lors du développement de notre algorithme de calibration.

## 2.2 Calibration par la variation de l'accélération horizontale (axe Y)

Afin d'éviter la déviation vers la gauche et le drift, nous avons décidé d'adopter la même approche que celle présentée dans la section 2.1.1, mais cette fois en nous appuyant sur les valeurs de l'accélération horizontale (sur l'axe Y). Nous avons également choisi de tester cette méthode pour observer si elle entraîne des modifications dans le comportement du robot après la calibration.

Contrairement à l'approche basée sur les données des capteurs de vitesse angulaire, qui sont également bruitées, les données provenant de l'accélération horizontale s'avèrent encore plus bruitées, même après l'application du filtre de Kalman. Par exemple, lorsque le robot est à l'arrêt, les valeurs filtrées ne sont pas nulles, mais présentent une incertitude d'environ  $\pm 5$ .

Concernant les résultats, sur notre robot de test, qui a une forte tendance à dévier vers la gauche, les valeurs du moteur gauche augmentent de manière incohérente avec la réalité. Comme indiqué dans la section 2, le robot a en effet tendance à se diriger vers la gauche, même lorsque les valeurs du moteur droit sont plus élevées. Par conséquent, cette approche ne s'avère pas pertinente pour améliorer l'algorithme de calibration.

## 2.3 Calibration en minimisant les résidus

Le code combine l'utilisation de l'Extended Kalman Filter (EKF) avec une boucle de calibration adaptative pour corriger la trajectoire d'un robot en temps réel. Les résidus calculés à chaque itération permettent d'ajuster dynamiquement les paramètres, optimisant ainsi la précision des estimations et la stabilité de la calibration.

Cette fonction ajuste les matrices de bruit ( $Q_k$  et  $R_k$ ) en fonction des résidus calculés pendant le filtrage :

- **Résidus élevés** : Si les erreurs moyennes sont grandes, les bruits de processus et de mesure sont augmentés.
- **Résidus faibles** : Si les erreurs sont faibles, les valeurs de bruit sont diminuées, améliorant ainsi la réactivité.

Lors de nos campagnes de tests, nous avons observé des résultats très variables malgré l'utilisation des mêmes paramètres. Nous avons réalisé plusieurs expérimentations et constaté les mêmes comportements que ceux décrits dans la section 2.1.1. Pourtant, cette partie de notre code visait à améliorer la précision de l'élimination du bruit effectuée par le filtre de Kalman. Cependant, les différences observées n'ont pas été significatives.

## 2.4 Observations sur nos expériences

En conséquence, les approches proposées n'ont pas permis au robot de se déplacer toujours en ligne droite lors de nos tests. À partir de nos observations, plusieurs raisons principales expliquent ces résultats :

1. Nous utilisons les valeurs du gyroscope sur l'axe z. Ces valeurs sont bruitées à cause des vibrations des moteurs. De plus, l'orientation du robot introduit un biais dans ces valeurs. Même si le robot reste droit avec les moteurs allumés, les valeurs de gyroZ ne sont jamais égales à 0.
2. Le moteur gauche est plus puissant que le moteur droit. Ce phénomène pousse le robot à dévier vers la gauche. Nous avons tenté de maximiser la puissance du moteur droit, mais le robot a continué à avoir tendance à dévier vers la droite.
3. Parfois, pendant la calibration, le moteur gauche s'allume avant le moteur droit, ce qui entraîne une certaine incertitude dans les données récupérées par les capteurs.
4. Durant la calibration, en raison des vibrations des moteurs, le robot peut tout de même changer de position dans son exosquelette, ce qui peut générer des points sans contact. Ces points sont cruciaux pour que les vibrations des moteurs produisent un déplacement correct. Par conséquent, le robot doit être solidement fixé dans son exosquelette (par exemple, avec de la pâte à fixe) afin d'augmenter les chances de réussite de la calibration.

Nos expérimentations ont mis en évidence trois types de comportements principaux : une calibration parfaite rare, une déviation fréquente vers la gauche, et un phénomène de dérive latérale. Ces problèmes sont exacerbés par le bruit des capteurs, en particulier sur l'axe Z, et les asymétries mécaniques du robot.

Pour de futures recherches, il sera nécessaire de mieux compenser les asymétries de l'exosquelette et de réduire l'impact du bruit des capteurs.

## 3 Choix de l'algorithme de calibration : expériences et améliorations

Nous avons donc décidé d'étudier plus en détail l'approche présentée dans la section 2.1.1, car elle a donné les meilleurs résultats lors de notre première série de tests, qui nous a permis de faire un choix entre les différentes méthodes de calibration.

### 3.1 Expériences

En ce qui concerne nos expériences, comme nous l'avons constaté lors de nos tests dans la section 2.1.3, l'augmentation du temps d'un essai (`try_duration` de la fonction `pogobot_calibrate`) peut influencer positivement la calibration de notre robot, car le filtre de Kalman récupère des données sur une plage plus large de temps, ce qui lui permet d'améliorer sa précision, tout en restant bruité.

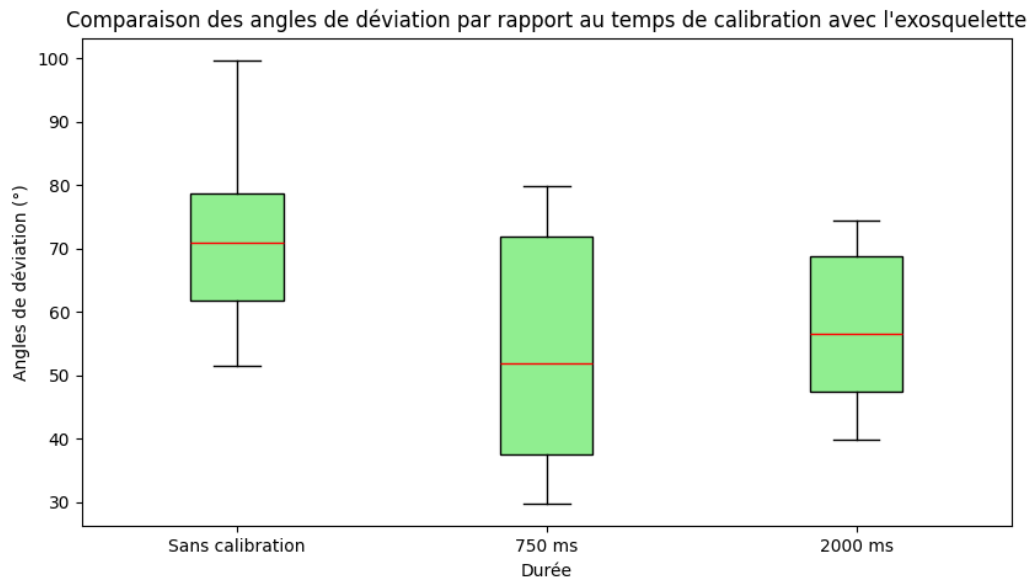
Pour ce faire, nous avons effectué 10 calibrations pour chaque durée : 750 ms, 2 s, et pour chaque type de squelette : exosquelette et brosses à dents. Ensuite, après la calibration, nous avons laissé le robot tourner pendant 10 secondes, puis nous avons mesuré son angle de déviation par rapport à sa position initiale après les 10 secondes. Les mesures ont été faites à l'aide du logiciel [Rhino](#), qui est un outil très précis permettant de mesurer des distances et des angles à partir des images données. Une illustration succincte de la méthode utilisée pour calculer les angles de déviation et les distances parcourues est fournie en annexe dans les figures 5 et 6.

Comme nous pouvons le constater dans la figure 3a, dans le cas de l'exosquelette, un robot calibré parvient à réduire ses angles de déviation par rapport à un robot sans calibration, quel que soit le temps d'essai. Nous observons également que les essais d'une durée de 2 secondes sont légèrement plus stables que ceux de 750 millisecondes, qui présentent davantage de variations. Cependant, la moyenne reste globalement similaire pour les deux durées.

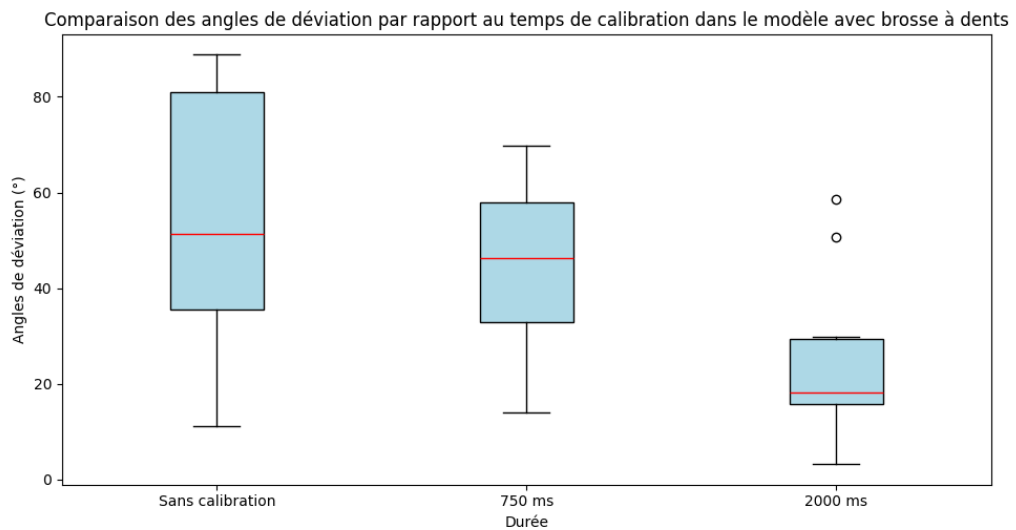
Dans le modèle avec des brosses à dents, figure 3b, nous observons le même comportement que pour l'exosquelette concernant un robot calibré ou non. Cela montre que notre algorithme permet d'améliorer le comportement du robot. Contrairement à l'exosquelette, les angles de déviation sont plus faibles dans tous les cas, ce qui indique que le modèle avec des brosses à dents est mieux adapté au pogobot. Cependant, dans ce modèle, une différence apparaît entre une calibration réalisée avec une durée d'essai de 750 millisecondes et celle de 2 secondes. Dans le cas de 2 secondes, nous constatons une amélioration des angles de déviation. Par conséquent, nous recommandons de réaliser la calibration avec une durée d'essai (`try_duration`) de 2 secondes.

Concernant les distances, dans le cas de l'exosquelette (figure 4a), nous observons que les distances varient également d'un run à l'autre, en particulier pour les durées de 750 millisecondes et de 2 secondes. Cependant, les moyennes convergent autour d'une distance parcourue similaire, qui, dans le cas de l'exosquelette, est d'environ 3,5 cm. Pour ce qui est des distances, nous avons choisi de ne pas inclure les mesures sans calibration en raison des faibles performances du Pogobot, qui a tendance à tourner sur lui-même dans les expérimentations avec l'exosquelette.





(a) Angles de déviation en fonction du temps d'un essai sur l'exosquelette



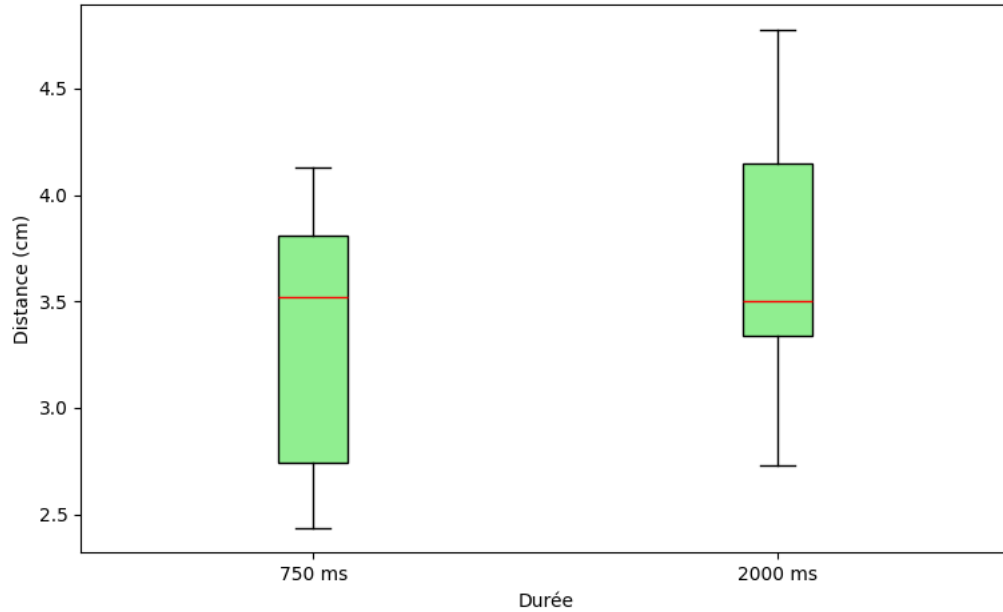
(b) Angles de déviation en fonction du temps d'un essai sur le modèle avec des brosses à dents

FIGURE 3 – Comparaison des angles de déviation par rapport au temps d'un essai

Dans le cas du modèle avec des brosses à dents (figure 4b), nous constatons également une différence dans les distances parcourues. Cependant, en prenant en compte les deux points situés autour de 10 cm et 8 cm, nous obtenons une distance moyenne similaire. Contrairement à l'exosquelette, ce modèle parvient à se déplacer beaucoup plus efficacement au cours des 10 secondes de test.

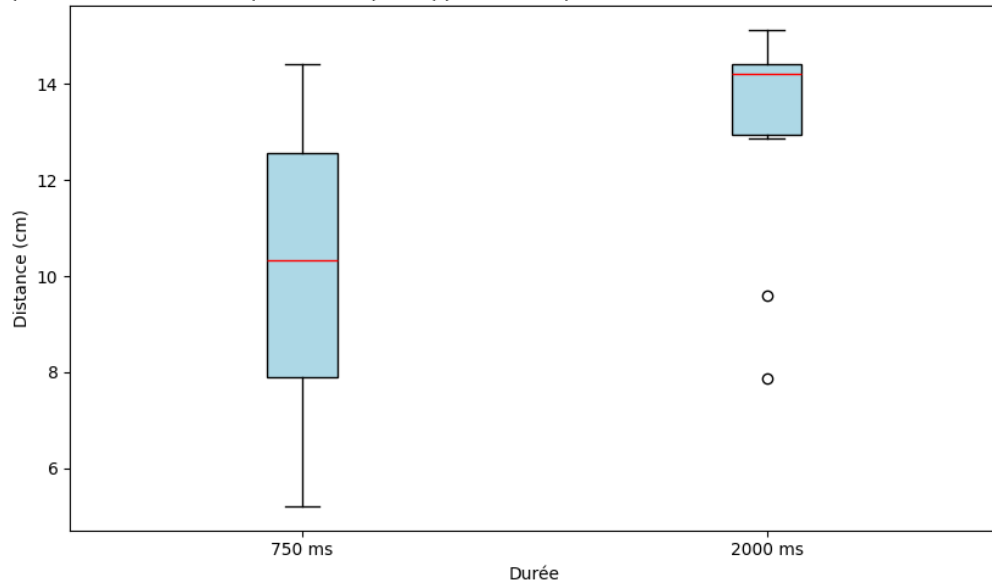


Comparaison des distances parcourues par rapport au temps de calibration avec l'exosquelette



(a) Distance parcourue en fonction du temps d'un essai sur l'exosquelette

Comparaison des distances parcourues par rapport au temps de calibration dans le modèle avec brosse à dents



(b) Distance parcourue en fonction du temps d'un essai sur le modèle avec des brosses à dents

FIGURE 4 – Comparaison des distances parcourues par rapport au temps d'un essai

## 4 Code de démonstration

Dans ce code, nous avons implémenté une série de fonctions permettant de contrôler les mouvements du pogobot une fois qu'il est calibré. Ces fonctions permettent d'exécuter différents types de déplacements, comme avancer en ligne droite, tourner légèrement ou fortement à gauche ou à droite, ainsi que s'arrêter. Chaque type de mouvement est accompagné d'un indicateur visuel sous forme de changement de couleur des LEDs, facilitant le suivi et l'analyse du comportement du robot. Les durées des mouvements sont uniformément fixées à 5 secondes, et nous avons développé une fonction personnalisée de temporisation (`pogobot_delay_ms`) pour assurer un contrôle précis des délais.

L'ensemble des mouvements est coordonné dans la fonction `demo`, où nous avons défini une boucle infinie qui enchaîne ces déplacements avec des pauses intermédiaires pour synchroniser les transitions. Afin d'adapter dynamiquement les puissances des moteurs, nous avons également conçu des fonctions permettant de récupérer les paramètres des moteurs gauche et droit depuis la mémoire du robot.

## 5 Conclusion

Ce projet a permis de développer et d'évaluer diverses approches pour la calibration automatique des Pogobots, en explorant les méthodes basées sur les variations de vitesse angulaire et d'accélération horizontale, ainsi que sur la minimisation des résidus. Malgré les défis liés aux bruits des capteurs et aux asymétries mécaniques des robots, certaines améliorations significatives ont été obtenues, notamment en ajustant la durée des essais de calibration.

L'approche utilisant des essais de 2 secondes a démontré une meilleure stabilité et une réduction des angles de déviation par rapport à des essais plus courts. Cependant, le comportement des robots reste fortement influencé par les caractéristiques de leur squelette et par la qualité des données des capteurs. L'algorithme testé, bien qu'amélioré, nécessite encore des ajustements pour compenser les asymétries mécaniques et réduire davantage le bruit des mesures.

Pour des travaux futurs, il serait intéressant de se concentrer sur des solutions plus robustes, telles que l'intégration de modèles prédictifs plus avancés ou l'utilisation de capteurs supplémentaires pour mieux compenser les incertitudes. Ce projet constitue ainsi une étape prometteuse vers des Pogobots capables de mouvements plus précis et fiables, tout en offrant des pistes d'amélioration claires pour optimiser leurs performances.

## Annexe

Méthodes de mesure des angles et distances lors de nos campagnes de tests sur le choix de notre algorithme de calibration

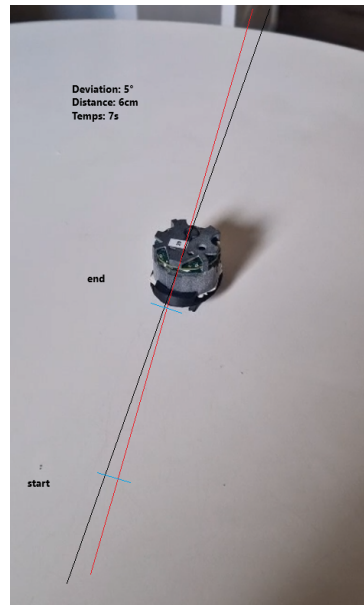


FIGURE 5 – Approche 2.1.1, 15 itérations, 750 ms par essai. Rouge : axe au début, Noir : axe après 7s.



FIGURE 6 – Approche 2.1.1, 15 itérations, 2 s par essai. Rouge : axe au début, Bleu : axe après 14s.