
RAPPORT DE PROJET

Projet d'électronique n°4 : Gyropode

Auteurs :

Laouïg
Robin

Enseignant :

M. LE GALL

Notre projet de "Gyrobot" consiste à développer un gyropode, un robot auto-équilibrant à deux roues, dans un contexte d'apprentissage et de mise en pratique des systèmes embarqués et des principes d'asservissement.

Utilisant le MPU6050 pour mesurer l'inclinaison et la vitesse, le robot doit pouvoir se stabiliser verticalement, maintenir un cap et une vitesse constante. Les défis techniques incluent l'intégration efficace des composants électroniques, la programmation de contrôleurs PID pour différents types de mouvements, et l'adaptation aux variations physiques des composants tels que les différences de puissance entre les roues.

Ce projet est mené dans le cadre de notre seconde année de classe préparatoire intégrée, dans l'optique de mettre en pratique nos connaissances théoriques acquises en ce début d'année.

Nous attestons que ce travail est original, qu'il est le fruit d'un travail commun au binôme et qu'il a été rédigé de manière autonome.

Paris, le 20/05/2024

Table des matières

I.	Étude documentaire	
II.	Objectif	4
III.	Glossaire	6
A.	Termes	
B.	Acronymes	
IV.	L'équipe	7
A.	Présentation de l'équipe	
B.	Organisation de l'équipe	
C.	Diagramme de GANTT	
V.	Contexte et problématique	9
A.	Contexte	
B.	Problématique	
C.	Spécifications techniques	
VI.	Conception	11
A.	Architecture fonctionnelle	
B.	Architecture matérielle	
C.	Architecture logicielle	
VII.	Développement	13
A.	ET1 : La centrale inertie	
B.	ET2 : Verticalité	
C.	ET3 : $t_{15\%}$ et ε^∞	
D.	ET4 : Ligne droite	
E.	ET5 : Vitesse	
VIII.	Tests et validation	20
A.	ET1 : La centrale inertie	
B.	ET2 : Verticalité	
C.	ET3 : $t_{15\%}$ et ε^∞	
D.	ET4 : Ligne droite	
E.	ET5 : Vitesse	
IX.	Bilan	27
A.	État d'avancement	
B.	Pertinence de la solution technique	
C.	Bilan sur le travail d'équipe	
X.	Sources	29
XI.	Annexes	29

I. Étude documentaire

A. Gyropode

Le Gyropode est une innovation majeure dans le monde des transports personnels. Ce véhicule électrique auto-équilibrant, imaginé par Dean Kamen et commercialisé par Segway Inc. du New Hampshire, représente une réponse moderne aux défis de mobilité urbaine. Silencieux, efficace et sans émissions polluantes, le Gyropode permet à son utilisateur de se déplacer en se penchant simplement dans la direction souhaitée, tout en maintenant l'équilibre automatiquement sur deux roues.

Inventé dans le but de révolutionner le transport personnel, Dean Kamen a souvent comparé son invention à l'impact que la voiture a eu après le cheval et la calèche. Introduit au grand public lors d'une apparition sur "Good Morning America", Kamen a présenté le Gyropode comme le premier transporteur humain auto-équilibrant. Ce dispositif intègre des technologies avancées telles que des servos électriques et des gyroscopes qui ajustent continuellement l'équilibre, permettant une expérience de glisse unique et intuitive.

Cependant, malgré ses avantages, le Gyropode n'a pas remplacé la voiture comme Kamen l'avait espéré. Sa vitesse limitée à 20 km/h et son autonomie de 24 km après six heures de charge ne lui permettent pas de concurrencer les véhicules traditionnels pour des trajets longs. Néanmoins, dans les milieux urbains, il offre une alternative pratique aux voitures, encombrantes et coûteuses à entretenir. Son faible encombrement lui permet de réduire les congestions et de naviguer facilement dans les foules, idéal pour les zones piétonnes denses comme les parcs d'attractions ou les entrepôts.

Les premières ventes n'ont pas été à la hauteur des attentes, avec seulement quelques dizaines de milliers d'unités écoulées depuis 2002. Le prix élevé a été un frein majeur, bien que la société ait tenté de le surmonter en introduisant des options de financement et de leasing. Convaincu du potentiel de son invention, Kamen a ciblé les agences gouvernementales et les grandes entreprises avant de s'attaquer au marché grand public.

En conclusion, le Gyropode, malgré un démarrage commercial modeste, a trouvé sa place dans des niches spécifiques où sa petite taille et sa maniabilité offrent des avantages significatifs. Il continue d'incarner l'innovation dans le domaine de la micro-mobilité urbaine, proposant une alternative intéressante pour les déplacements courts et écologiques en ville.

B. Fonctionnement de la centrale inertielle (MPU6050)

La MPU6050 est un système micro-électromécanique (MEMS) qui intègre un accéléromètre triaxial et un gyroscope triaxial. Ce capteur est essentiel pour mesurer l'accélération, la vitesse, l'orientation, le déplacement, et d'autres paramètres liés au mouvement d'un objet ou système.

Configuration des Broches du MPU6050

- **Vcc (Pin 1):** Fournit l'alimentation au module, pouvant varier de +3V à +5V, typiquement +5V.
- **Ground (Pin 2):** Connecté à la masse du système.
- **Serial Clock (SCL, Pin 3) et Serial Data (SDA, Pin 4):** Utilisés pour la communication I2C.
- **Auxiliary Serial Data (XDA, Pin 5) et Clock (XCL, Pin 6):** Facultatifs, pour interfaçer d'autres modules I2C.
- **AD0 (Pin 7):** Utilisé pour modifier l'adresse I2C si plusieurs MPU6050 sont utilisés.
- **Interrupt (INT, Pin 8):** Indique la disponibilité des données à lire.

Caractéristiques Techniques

- **Combinaison d'accéléromètre et gyroscope MEMS à 3 axes.**
- **Alimentation : 3-5V**
- **Communication : Protocole I2C**
- **Convertisseur ADC 16 bits intégré pour une haute précision**
- **Processeur de mouvement numérique intégré (DMP) pour une puissance de calcul élevée**
- **Capteur de température intégré**
- **Adresse IIC configurable**

Le MPU6050 est un capteur combinant à la fois un accéléromètre et un gyroscope, intégré dans un seul module. Ce dispositif est largement utilisé dans divers projets de robotique et d'électronique en raison de sa capacité à fournir des mesures précises du mouvement et de l'orientation. Voici une explication simplifiée de son fonctionnement, adaptée même pour ceux qui débutent dans ce domaine.

Fonctionnement du MPU6050

Le MPU6050 appartient à la catégorie des systèmes microélectromécaniques (MEMS), qui combinent des éléments mécaniques miniatures et des circuits électroniques sur un seul substrat de silicium. Le module se compose d'un accéléromètre triaxial et d'un gyroscope triaxial. Cela signifie qu'il peut mesurer à la fois l'accélération et la vitesse angulaire sur trois axes : X, Y et Z.

Accéléromètre

L'accéléromètre du MPU6050 mesure l'accélération linéaire. Autrement dit, il détecte le mouvement ou les changements de position dans l'espace. Cela est utile pour détecter la vitesse à laquelle un objet se déplace ou s'il est en train de tomber. Les données peuvent aider à comprendre comment un objet est orienté par rapport à la surface de la Terre.

Gyroscope

Le gyroscope mesure la vitesse angulaire, c'est-à-dire la rapidité avec laquelle un objet tourne autour d'un axe. Cette information est cruciale pour déterminer l'orientation précise d'un objet en mouvement, comme la direction dans laquelle il pointe à tout moment. C'est particulièrement utile dans les applications de navigation et de contrôle de la stabilité, comme les drones ou les robots équilibrés automatiquement.

Processeur de Mouvement Numérique (DMP)

Un aspect distinctif du MPU6050 est son Processeur de Mouvement Numérique (DMP) intégré. Le DMP peut exécuter des algorithmes complexes de traitement du signal pour fusionner les données de l'accéléromètre et du gyroscope. Cela libère le microcontrôleur principal du système de cette tâche gourmande en ressources, permettant ainsi de gérer d'autres tâches simultanément ou d'améliorer l'efficacité énergétique du système.

Applications

Les informations fournies par le MPU6050 sont utilisées pour de nombreuses applications pratiques. Par exemple, dans les smartphones, ces capteurs permettent de détecter l'orientation de l'écran (portrait ou paysage) et de réagir aux gestes de l'utilisateur, comme secouer ou incliner le téléphone. Dans le domaine de la robotique, ils aident à contrôler et à stabiliser les mouvements des robots. Pour les véhicules autonomes comme les drones, ils sont essentiels pour maintenir le vol stable et naviguer dans l'environnement.

En résumé, le MPU6050 est un outil extrêmement polyvalent pour tout projet nécessitant une mesure précise du mouvement et de l'orientation. Grâce à sa capacité à traiter des données complexes de manière autonome et son interface simple avec des microcontrôleurs via le protocole I2C, il reste un choix populaire parmi les amateurs et les professionnels de l'électronique et de la robotique.

C. Fonctionnement d'un servomoteur

Les servomoteurs sont des éléments essentiels dans le domaine de la robotique et des systèmes automatisés, offrant un contrôle précis de la position, de la vitesse et de l'accélération. Ils fonctionnent selon un principe de boucle fermée, ce qui est crucial pour atteindre la précision et la réactivité nécessaires dans de nombreuses applications, allant de l'automatisation industrielle aux gadgets pour loisirs.

Fonctionnement en Boucle Fermée

Un servomoteur comprend un moteur électrique, un capteur de retour et une unité de contrôle électronique. Le moteur peut être à courant alternatif ou continu et peut intégrer des technologies avec ou sans balais. Le capteur fournit un retour en temps réel sur des paramètres tels que la position, la vitesse ou le courant, indispensables pour le contrôle précis du moteur.

Le fonctionnement de base d'un servomoteur repose sur le suivi continu et la correction de la position ou de la vitesse du moteur par rapport à une commande. Le capteur mesure la position actuelle et transmet ces informations à l'unité de contrôle, qui ajuste alors la puissance fournie au moteur pour atteindre la position ou la vitesse désirée. Ce processus se répète plusieurs fois par seconde, permettant un ajustement précis et rapide des actions du moteur.

Composantes d'un Servomoteur

Les servomoteurs sont composés de trois éléments principaux :

- **Moteur** : Le cœur du servomoteur, qui convertit l'énergie électrique en énergie mécanique.
- **Capteur d'asservissement** : Permet de surveiller en continu le fonctionnement du moteur et de fournir des données de retour à l'unité de contrôle. Les capteurs utilisés peuvent varier (potentiomètres, encodeurs, résolveurs, etc.) selon les besoins de précision.
- **Électronique de contrôle** : Compare les données du capteur avec les consignes et ajuste l'alimentation du moteur pour corriger tout écart.

Exemple de Configuration

Dans un exemple simple, un servomoteur industriel pourrait utiliser un moteur à courant continu avec un tachymètre intégré. Le tachymètre génère une sortie proportionnelle à la vitesse du moteur. L'unité de contrôle ajuste la tension et le courant alimentant le moteur pour maintenir la vitesse souhaitée, basée sur la tension de retour du tachymètre. Cette configuration est un exemple de "boucle de vitesse fermée", visant à conserver une vitesse constante selon les paramètres de l'utilisateur.

Boucles Multiples pour Performance Optimale

Dans des systèmes plus complexes, plusieurs boucles peuvent être intégrées pour un contrôle plus raffiné, telles que les boucles de courant, de vitesse et de position, chacune destinée à ajuster un aspect spécifique du mouvement du moteur :

- **Boucle de courant** : Contrôle le courant pour ajuster le couple produit par le moteur.
- **Boucle de vitesse** : Ajuste la vitesse du moteur pour suivre la consigne de vitesse.
- **Boucle de position** : Gère la position finale du moteur, souvent utilisée dans les applications où la précision de la position est critique.

Grâce à leur fonctionnement en boucle fermée, les servomoteurs offrent des performances très précises et réactives, indispensables dans les systèmes automatisés modernes. Que ce soit dans les robots, les voitures autonomes ou les systèmes de production industrielle, ils jouent un rôle clé en assurant que le mouvement et la position des composants soient exactement conformes aux exigences des applications. Leur compréhension et leur maîtrise sont essentielles pour les ingénieurs et techniciens œuvrant dans l'automatisation et la robotique.

II. Objectif

Ce rapport retrace notre parcours avec le projet Gyrobot, où nous avons allié théorie et pratique pour façonnez un gyropode auto-équilibrant. Vous y découvrirez notre expérience avec la centrale inertie MPU6050, utilisée pour mesurer l'inclinaison et la vitesse du robot, et notre travail avec les contrôleurs PID qui nous ont permis de naviguer entre différents mouvements robotiques.

Nous avons aussi tenu à partager les défis rencontrés, tels que l'intégration des composants électroniques et les ajustements nécessaires face aux différences de puissance entre les roues.

À travers les différentes sections du rapport, nous dévoilons les étapes clés de notre projet, de la conception à la validation finale, illustrées par des schémas et des algorithmes. Ces éléments visuels accompagnent nos résultats expérimentaux, reflétant les avancées et les performances du système. Ce document est un partage de notre aventure d'apprentissage, avec ses hauts et ses bas, dans l'espoir d'inspirer et d'informer.

III. Glossaire

D. Termes

Terme	Définition
Gyropode	Type de transport personnel à deux roues qui utilise des gyroscopes et des capteurs pour maintenir l'équilibre automatiquement, permettant à l'utilisateur de se déplacer en se penchant dans la direction souhaitée.
Centrale inertielle (IMU)	Dispositif électronique qui mesure et rapporte la vitesse, l'orientation et les forces gravitationnelles d'un objet, en utilisant une combinaison d'accéléromètres et de gyroscopes.
Contrôle PID (Proportionnel, Intégral, Dérivatif)	Méthode de contrôle de boucle fermée utilisée dans les systèmes automatisés pour réguler continuellement une variable de processus en calculant l'écart entre une valeur désirée et une valeur mesurée, et en appliquant une correction basée sur des termes proportionnels, intégraux, et dérivatifs.
Communication I2C	Protocole de communication série utilisé pour connecter des périphériques basse vitesse comme des microcontrôleurs, des radios, et des capteurs à une carte mère, un microcontrôleur ou un autre système numérique.
Modulation de largeur d'impulsion (PWM)	Technique de modulation utilisée pour contrôler la quantité d'énergie fournie à un dispositif électronique en décomposant le signal entre complètement allumé et complètement éteint à différentes fréquences.
Servomoteur	Type de moteur qui permet un contrôle précis de la position angulaire, de la vitesse et de l'accélération. Dans le cadre de ce projet, ils sont utilisés pour ajuster la position du robot afin de maintenir l'équilibre.

E. Acronymes

Acronyme	Signification	Explication
I2C	(Inter-Integrated Circuit) Circuit Inter-Intégré	Circuit Inter-Intégré Protocole de communication série
MPU	Unité de Traitement de Mouvement	Composant souvent présent dans les systèmes de capteurs inertIELS comme la MPU6050, utilisé pour traiter les données de mouvement.

IV. L'équipe

A. Présentation de l'équipe



Nous sommes Laouïg ELEOUET et Robin QUERIAUX, et former une équipe de deux a été une décision mûrement réfléchie pour tirer le meilleur de notre collaboration sur ce projet.

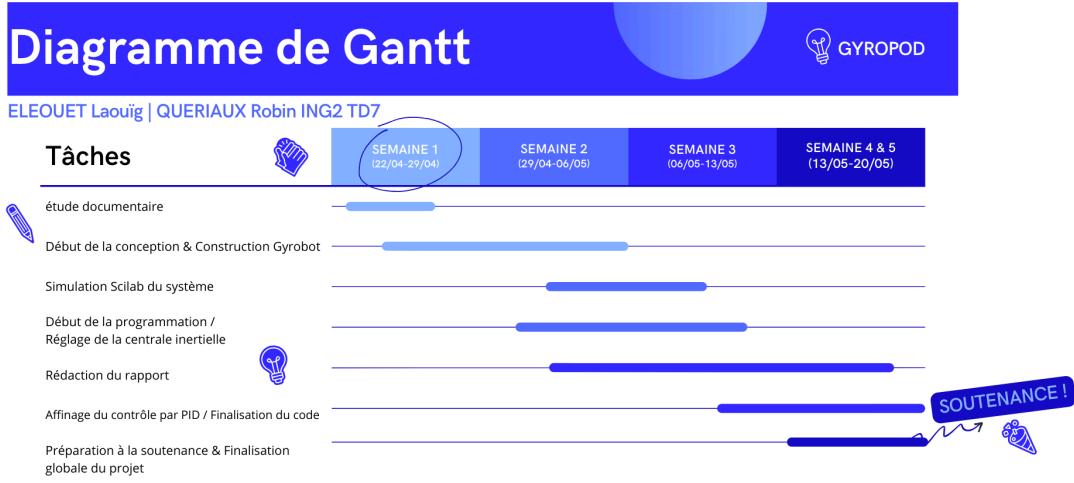
Laouïg s'est immergé dans la partie technique, prenant en main la conception et les tests de calibrage. Il a également géré une bonne partie des simulations. Sa précision et son souci du détail ont été indispensables pour peaufiner notre prototype et assurer sa performance optimale.

Quant à moi, Robin, j'ai pris en charge l'étude documentaire et contribué aux simulations. J'ai veillé à ce que notre travail soit bien fondé et documenté, en m'appuyant sur des recherches approfondies pour éclairer nos choix techniques et stratégiques. Ma passion pour creuser au-delà des évidences a souvent aidé à résoudre des problèmes complexes et à améliorer notre compréhension du projet.

B. Organisation de l'équipe

Puisque nous n'avions pas toujours la possibilité de nous retrouver physiquement, nous avons divisé les tâches et organisé des réunions régulières pour discuter de nos avancées. Ces moments de partage ont été essentiels : ils nous ont permis de rester en phase et de surmonter ensemble les défis, renforçant notre esprit d'équipe et notre engagement envers le succès du Gyrobot.

C. Diagramme de GANTT



V. Contexte et problématique

A. Contexte

Le contexte économique et sociétal de notre projet Gyrobot s'inscrit dans une ère où l'automatisation et la mobilité personnelle prennent une importance croissante. Les innovations dans les technologies de transport personnel, telles que les gyropodes, répondent à une demande croissante pour des solutions de mobilité urbaine plus flexibles et durables. Cela est particulièrement pertinent dans les zones urbaines densément peuplées où la congestion et la pollution sont des problèmes majeurs. Les gyropodes, en offrant une alternative écologique et compacte aux véhicules traditionnels, peuvent contribuer à réduire ces impacts tout en offrant une manière pratique et efficace de naviguer dans les espaces urbains.

La technologie du gyropode, quant à elle, est née de l'ambition de révolutionner le transport personnel. Inventée au début des années 2000 par Dean Kamen et commercialisée sous la marque Segway, cette technologie utilise des gyroscopes et des systèmes de contrôle avancés pour maintenir l'équilibre automatiquement, permettant à l'utilisateur de se déplacer en se penchant simplement dans la direction souhaitée. Depuis son introduction, la technologie du gyropode a évolué avec des améliorations continues en termes de sécurité, de maniabilité, et d'efficacité énergétique, rendant ces appareils plus accessibles et plus attractifs pour un usage quotidien.

Dans notre projet Gyrobot, nous avons adopté cette base technologique pour développer un prototype qui intègre des innovations en termes de contrôle et de stabilité, en visant une application pédagogique et de recherche qui pourrait également servir de tremplin pour de futures applications commerciales. Notre objectif était de pousser plus loin l'automatisation et l'adaptabilité de ces systèmes dans divers contextes d'utilisation, tout en tenant compte des enjeux économiques et sociaux actuels.

B. Problématique

Dans le cadre de notre projet Gyrobot, nous avons abordé la problématique de stabilisation et de contrôle d'un gyropode. Ce type de robot, qui se caractérise par sa capacité à maintenir une position verticale tout en se déplaçant, pose un défi significatif en termes d'asservissement et de correction dans le domaine de la robotique et de l'automatique.

Pour relever ce défi, notre gyropode utilise une centrale inertielle MPU6050 pour mesurer l'angle d'inclinaison et d'autres paramètres critiques. Les actions correctrices sont exécutées par des servomoteurs à rotation continue, contrôlés par une carte Arduino Nano Atmega328P qui traite les données de la centrale et ajuste le comportement des moteurs. Ce système permet au robot de conserver son cap et sa vitesse, tout en restant stable.

Ce projet s'inscrit également dans un contexte pédagogique enrichissant. Il nous a permis d'appliquer des connaissances théoriques en asservissement numérique PID, tout en développant des compétences pratiques en documentation technique et gestion de projet suivant un cycle en V. Par ce projet, nous avons non seulement renforcé notre compréhension théorique des systèmes dynamiques, mais avons également acquis des compétences clés en matière de gestion et réalisation de projets techniques complexes.

C. Spécifications techniques

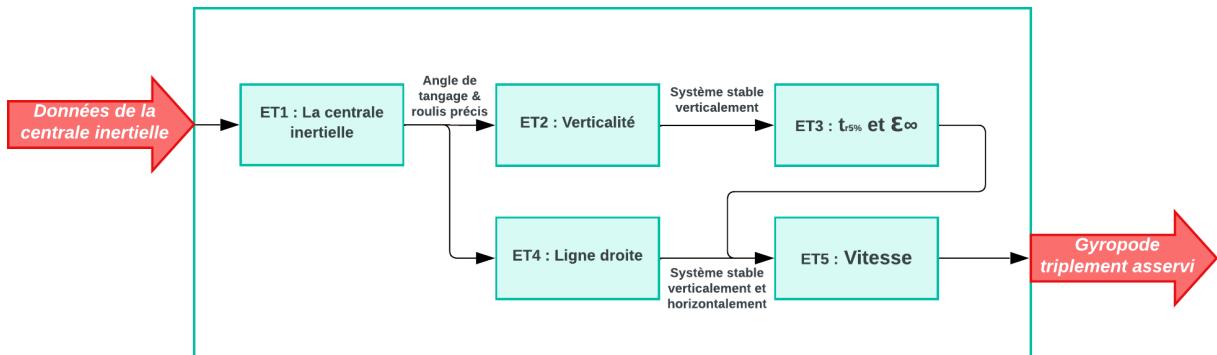
Pour le projet Gyrobot, nous avons défini plusieurs spécifications techniques pour garantir que le robot réponde à nos attentes en termes de performance, de stabilité, et de fonctionnalité. Voici les spécifications techniques principales :

1. **Stabilité:** Le robot doit pouvoir se maintenir à la verticale avec une amplitude d'oscillation n'excédant pas $\pm 15^\circ$ durant son fonctionnement. (ET1)
2. **Navigation:** Capacité de se déplacer en ligne droite sur au moins 80 cm dans un couloir de 25 cm de large, tout en maintenant une trajectoire précise. (ET2&4)
3. **Contrôle de vitesse:** Le robot doit pouvoir maintenir une vitesse constante avec une variation maximale de $\pm 10\%$ pendant au moins 10 secondes. (ET5)
4. **Temps de réponse et précision:** Le temps de réponse à une commande ne doit pas excéder 1 ms, et l'erreur statique en régime permanent doit être inférieure à 8%. (ET3)
5. **Mesure d'angle:** Utilisation de la centrale inertielle MPU6050 pour mesurer les angles de tangage avec une précision maximale de 2% d'erreur. (ET3)
6. **Alimentation:** Le robot doit être alimenté par une connexion USB, fournissant une source d'énergie stable et facile à gérer.

Ces spécifications ont été établies pour assurer que notre robot gyropode fonctionne de manière efficace et sûre, répondant à la fois aux besoins et aux contraintes techniques du projet.

VI. Conception

A. Architecture fonctionnelle



B. Architecture matérielle

Pour notre projet Gyrobot, nous avons utilisé des composants clés spécifiquement choisis pour leur fiabilité et leur performance dans la construction d'un robot gyropode auto-équilibrant.

1. **Centrale inertie MPU6050** : Utilisée pour mesurer avec précision l'angle d'inclinaison et la vitesse du robot, essentielle pour son équilibrage automatique.
2. **Arduino Nano ATMEGA328P** : Sert de contrôleur principal, gérant les signaux de la centrale inertie et exécutant les algorithmes de contrôle pour ajuster les mouvements des moteurs.
3. **Servomoteurs à rotation complète FS90R** : Permet un contrôle précis et continu du mouvement, essentiel pour maintenir l'équilibre du gyropode et gérer sa navigation.

- La **centrale inertie** est connectée à l'**Arduino** via I2C, permettant une transmission rapide et efficace des données.
- L'**Arduino** pilote les **servomoteurs** grâce à des signaux PWM, optimisant la réactivité du système.
- L'alimentation via USB offre une solution simple et sécurisée pour l'énergie, facilitant les tests et l'entretien.

Ces choix de conception garantissent que notre Gyrobot est stable, réactif, et capable de manœuvres précises, répondant aux objectifs et exigences techniques de notre projet.

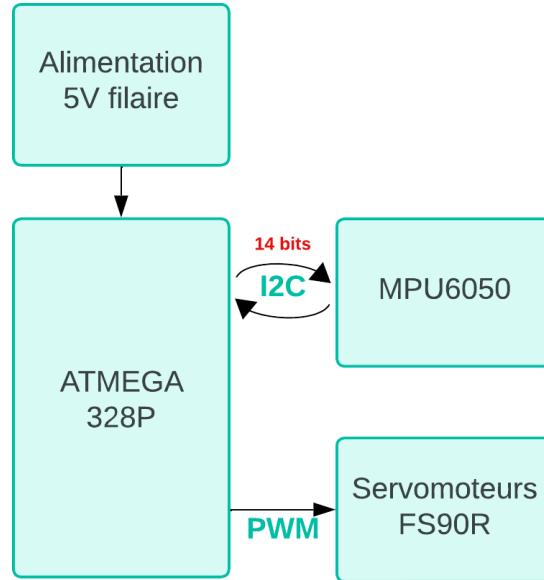


Figure : Architecture matérielle du GYROBOT

C. Architecture logicielle

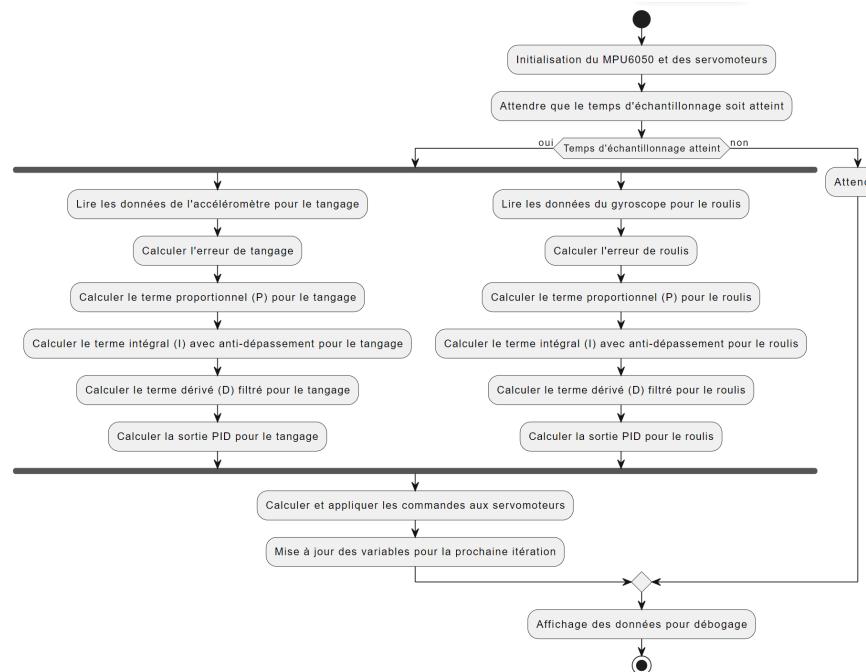


Figure : Architecture matérielle du GYROBOT

VII. Développement

A. ET1 : La centrale inertie

Pour donner vie à notre projet Gyrobot, nous avons intégré un élément clé : la centrale inertie MPU6050. Ce composant est essentiel pour mesurer précisément l'inclinaison du robot, ce qui est indispensable pour son auto-équilibrage. Jetons un coup d'œil plus approfondi sur la manière dont nous avons programmé ce capteur pour qu'il réponde à nos besoins.

Calcul des Angles:

Nous calculons l'angle de tangage `x` en utilisant la fonction `atan2()` pour l'axe Y par rapport à Z, ce qui donne un aperçu de la façon dont le robot s'incline vers l'avant ou l'arrière. La correction de -90 degrés ajuste l'angle pour qu'il soit centré à zéro lorsque le robot est droit.

Un problème matériel notable a été lié au câble USB que nous utilisions pour connecter l'Arduino Nano à notre ordinateur. Nous avons rencontré de fréquentes interruptions de connexion et des transferts de données erratiques. Après de nombreuses tentatives de dépannage, nous avons déterminé que le câble USB était défectueux. Nous avons dû en commander un nouveau sur internet, ce qui a retardé notre progression mais a finalement résolu le problème de connectivité.

Lors de nos premiers tests, nous avons constaté que les données fournies par le MPU6050 étaient sujettes à des fluctuations importantes. Cela a conduit à une instabilité dans l'auto-équilibrage du robot. Pour résoudre ce problème, nous avons implémenté un filtre passe-bas logiciel pour lisser les données brutes du capteur. Cela a considérablement amélioré la précision des mesures d'angle de tangage.

La calibration du MPU6050 a été un processus délicat. Utiliser un rapporteur physique pour ajuster les valeurs de minVal et maxVal était imprécis et chronophage. Nous avons donc développé un algorithme de calibration automatique qui enregistre les valeurs extrêmes pendant que le robot est manipulé dans différentes positions. Cette méthode a non seulement accéléré la calibration mais a aussi augmenté la précision des mesures.

Pour éviter les interruptions intempestives qui perturbent la lecture des capteurs et le contrôle des moteurs, nous avons opté pour une programmation sans interruption en utilisant la fonction millis() d'Arduino. Au lieu d'utiliser des délais bloquants ou des interruptions, nous avons écrit un programme qui exécute des tâches à intervalles réguliers en vérifiant le temps écoulé depuis le dernier cycle :

Cette approche a permis une exécution fluide et réactive du code, réduisant les délais et améliorant la stabilité du robot en temps réel.

L'aspect le plus stimulant a été d'assurer que les données du MPU6050 soient précises et fiables. Les fluctuations et les erreurs de mesure peuvent dramatiquement affecter la performance du robot. Pour contrer cela, nous avons passé du temps à implémenter un filtre passe bas et à calibrer le système, ajustant les valeurs de `minVal` et `maxVal` pour correspondre étroitement aux angles réels mesurés avec un rapporteur physique.

Ce travail pratique avec la MPU6050 a renforcé notre compréhension de la communication I2C et du traitement des données de capteurs, mais a aussi illustré l'importance de la précision dans la conception de systèmes robotiques. En fin de compte, ces compétences et ces expériences se traduiront directement par une meilleure préparation pour nos futures carrières en ingénierie, où les systèmes intégrés et la robotique continuent de jouer des rôles pivots.

B. ET2 : Verticalité

Pour notre projet Gyrobot, assurer la verticalité du robot est un élément crucial de sa stabilité. Voici un aperçu détaillé de la manière dont nous avons abordé et développé ce module, en couvrant les calculs théoriques, les choix techniques, et le concept clé derrière sa fabrication.

Théorie et Calculs

Le maintien de la verticalité repose sur l'asservissement de l'angle de tangage mesuré par la centrale inertie MPU6050. L'objectif est de maintenir cet angle proche de zéro pour que le robot reste droit. Pour cela, nous avons implémenté un contrôleur PID (Proportionnel, Intégral, Dérivé).

Calcul de l'Angle de Tangage :

L'angle de tangage θ est calculé en utilisant les données des accéléromètres de l'axe Y et Z du MPU6050. La formule utilisée est :

$$\theta = \text{atan2}(A_y, A_z)$$

Cette formule donne l'angle par rapport à la verticale, essentiel pour l'équilibrage du robot.

Implémentation du PID :

Le contrôleur PID ajuste la vitesse des servomoteurs pour corriger toute déviation de l'angle de tangage. Les gains Kp, Ki, et Kd sont ajustés empiriquement pour obtenir une réponse rapide et stable :

$$\text{Sortie PID} = K_p \cdot \text{erreur} + K_i \cdot \int \text{erreur} dt + K_d \cdot \frac{d(\text{erreur})}{dt}$$

Choix Techniques

Calibration du MPU6050 :

La précision et la qualité des valeurs du capteur est cruciale. Nous avons calibré le MPU6050 pour minimiser les erreurs de mesure. Les valeurs minimales et maximales des accéléromètres ont été ajustées en utilisant un rapporteur physique pour correspondre étroitement aux angles réels.

Filtrage des Données :

Pour réduire le bruit des données du capteur, nous avons appliqué un filtre passe-bas. Cela améliore la précision des mesures et la stabilité du PID.

Visualisation des Données

Moniteur Série Arduino :

Les valeurs de l'angle de tangage sont envoyées en temps réel au moniteur série de l'IDE Arduino. Cette visualisation nous a permis de déboguer et d'ajuster les paramètres PID en observant la réponse du robot aux perturbations.

Le développement de ce module nous a permis de comprendre en profondeur le fonctionnement et l'importance des systèmes d'asservissement et des capteurs inertIELS. La mise en œuvre du PID pour maintenir la verticalité du robot est un exemple concret de l'application des théories de contrôle automatique. Ces compétences sont directement transférables à d'autres projets en robotique et en systèmes intégrés, renforçant notre préparation pour des carrières en ingénierie.

Ce module sur la verticalité a démontré l'importance de la précision des mesures et de l'ajustement des paramètres pour la stabilité des systèmes robotiques, offrant une expérience pratique précieuse dans notre formation.

C. ET3 : $tr5\%$ et ε^∞

Théorie et Calculs

Temps de Réponse à 5% ($tr5\%$) :

Le temps de réponse à 5% $tr5\%$ est le temps nécessaire pour que la sortie du système atteigne et reste dans une bande de $\pm 5\%$ de la valeur finale après une entrée en échelon. Cet indicateur est crucial pour évaluer la rapidité et l'efficacité du système de contrôle du robot.

Pour le calculer, nous avons effectué une analyse temporelle de la réponse du système à une consigne donnée. Le PID joue un rôle crucial ici, car il doit être bien ajusté pour minimiser le temps de réponse tout en évitant des oscillations excessives.

Erreur Statique (ε^∞) :

L'erreur statique ε^∞ représente la différence permanente entre la consigne et la sortie du système en régime permanent. Une faible erreur statique est essentielle pour que le robot maintienne une position stable et précise.

Elle est déterminée par la composante proportionnelle et intégrale du PID. En régime permanent, l'intégrale de l'erreur doit compenser toute déviation résiduelle, ce qui permet d'atteindre une erreur statique minimale.

Choix Techniques

Implémentation du PID :

Pour atteindre les performances souhaitées en termes de $tr5\%$ et ε^∞ sur le roulis du gyrobot, nous avons soigneusement ajusté les gains K_p , K_i , et K_d . Voici le rôle de chaque composante :

- K_p (Proportionnel) : Augmente la réactivité du système, mais peut induire des oscillations.
- K_i (Integral) : Élimine l'erreur statique en intégrant l'erreur au fil du temps.
- K_d (Dérivé) : Atténue les oscillations en considérant la vitesse de changement de l'erreur.

Calibration et Ajustements

Optimisation du PID :

Pour optimiser le PID, nous avons réalisé plusieurs tests en ajustant les gains et observant les courbes de réponse du système. Nous avons cherché à minimiser **tr5%** tout en assurant que ϵ^∞ soit proche de zéro.

Méthode de Ziegler-Nichols :

Nous avons utilisé la méthode de Ziegler-Nichols pour une première approximation des paramètres du PID. Cette méthode implique d'augmenter **K_p** jusqu'à ce que le système oscille en régime permanent, puis de calculer **K_i** et **K_d** en fonction de cette valeur critique.

Tests et Résultats

Mesure du Temps de Réponse :

Nous avons effectué des acquisitions temporelles de la réponse en échelon du système. En traçant ces courbes, nous avons mesuré **tr5%** en déterminant le moment où la réponse atteint 95% de sa valeur finale et reste dans cette bande.

Calcul de l'Erreur Statique :

Pour déterminer ϵ^∞ , nous avons observé la sortie du système en régime permanent par rapport à la consigne. Nous avons ajusté le gain intégral **K_i** pour minimiser cette erreur, tout en gardant le système stable.

Le développement du module pour **tr5%** et ϵ^∞ a mis en lumière l'importance d'un réglage précis des contrôleurs PID pour atteindre des performances optimales. La compréhension et l'application des concepts de réponse temporelle et d'erreur statique sont essentielles dans la conception de systèmes d'asservissement efficaces.

D. ET4 : Ligne droite

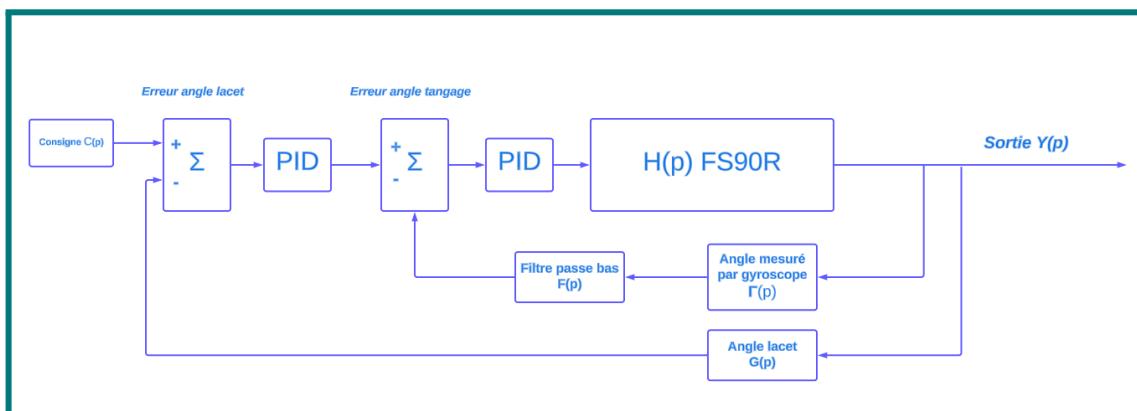


Schéma bloc du système doublement asservi (angle de tangage & lacet)

Théorie et Calculs

Maintien de la Ligne Droite :

Pour que le Gyrobot se déplace en ligne droite, il est essentiel de contrôler précisément la vitesse et la direction de chaque roue. Cela repose sur l'utilisation d'un PID pour chaque servomoteur, assurant que les deux moteurs tournent à la même vitesse et que le robot ne dévie pas de sa trajectoire.

Calculs de Correction :

La ligne droite est maintenue en surveillant l'angle de cap et les vitesses des roues. Le capteur MPU6050 fournit des données sur l'angle de cap (direction). La correction est calculée en ajustant la vitesse relative des servomoteurs :

$$\Delta v = K_p * erreur_{cap} + K_i * \int erreur_{cap} dt + K_d * \frac{d(erreur_{cap})}{dt}$$

Où Δv est l'ajustement de la vitesse des moteurs, et $erreur_{cap}$ est la différence entre le cap actuel et le cap désiré (idéalement zéro).

Choix Techniques

Utilisation du Capteur Inertielle MPU6050 :

Le MPU6050 mesure l'angle de cap en temps réel. Ces données sont essentielles pour détecter toute déviation de la trajectoire souhaitée. La communication avec l'Arduino Nano se fait via le protocole I2C, permettant une mise à jour rapide et précise des données.

Contrôle des Servomoteurs :

Les servomoteurs modifiés pour une rotation continue sont contrôlés en utilisant les valeurs calculées par les PID. Chaque moteur est asservi pour maintenir une vitesse constante, et des ajustements sont faits pour corriger toute déviation de la trajectoire.

Calibration et Ajustements

Initialisation et Calibration :

Pour garantir que le robot se déplace en ligne droite, nous avons initié les servomoteurs pour qu'ils aient des vitesses initiales égales. Ensuite, nous avons utilisé les données de pitch (tangage) fournies par le MPU6050 pour ajuster continuellement l'orientation du robot. En calculant l'angle de tangage autour de l'axe Y, nous avons pu corriger les déviations latérales et assurer que le robot maintienne une trajectoire rectiligne.

Optimisation des PID :

Les gains des PID pour chaque servomoteur ont été ajustés pour assurer une réponse rapide et stable. Nous avons utilisé des techniques comme la méthode de Ziegler-Nichols pour obtenir des valeurs initiales, puis affiné les gains par des tests empiriques.

Tests et Résultats

Test de Déplacement :

Nous avons effectué des tests en traçant des lignes droites sur une surface plane et en observant la capacité du robot à suivre ces lignes. Les données de cap et de vitesse ont été enregistrées pour analyser les performances du PID.

Correction des Déviations :

Toute déviation de la trajectoire a été corrigée en ajustant les PID, en particulier les gains proportionnels et dérivés, pour obtenir une trajectoire aussi rectiligne que possible.

Le développement du module pour le déplacement en ligne droite a démontré l'importance d'un contrôle précis des moteurs et de l'utilisation efficace des données de capteur. En ajustant soigneusement les PID pour chaque servomoteur, nous avons pu atteindre un déplacement stable et rectiligne, ce qui est essentiel pour la navigation et les tâches de précision.

E. ET5 : Vitesse

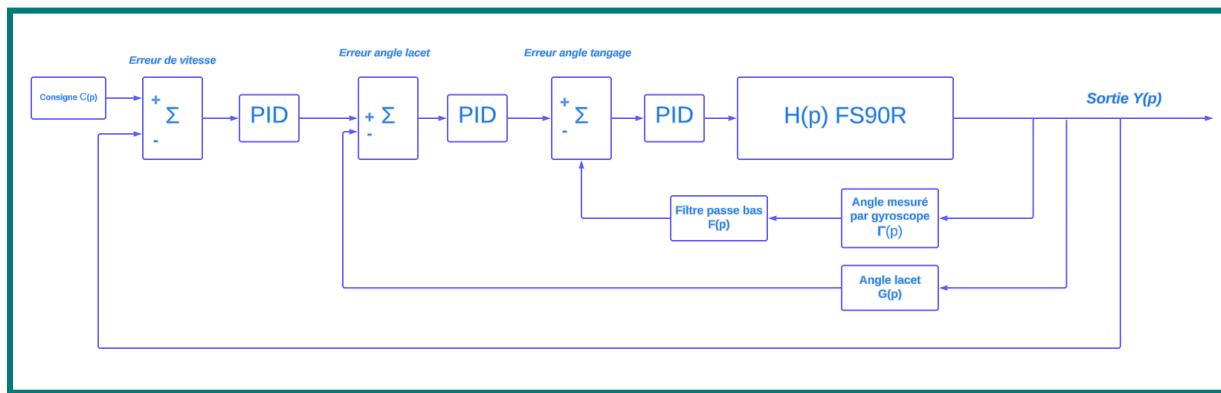


Schéma bloc du système triplement asservi (vitesse, angle de tangage & lacet)

Maintien de la Vitesse Constante :

Pour que le Gyrobot maintienne une vitesse constante, il est crucial de contrôler précisément les moteurs pour compenser toute variation due aux irrégularités de la surface ou aux différences de performance des moteurs. L'utilisation d'un contrôleur PID est essentielle pour ajuster la vitesse en temps réel.

Calculs de Correction :

La vitesse est contrôlée en surveillant la sortie du capteur MPU6050 pour mesurer l'accélération, puis en intégrant ces données pour obtenir la vitesse. Le PID ajuste la puissance envoyée aux moteurs pour maintenir la vitesse désirée.

$$v = K_p * \text{erreur}_\text{vitesse} + K_i * \int \text{erreur}_\text{vitesse} dt + K_d * \frac{d(\text{erreur}_\text{vitesse})}{dt}$$

Où v est l'ajustement de la vitesse des moteurs, et erreur_vitesse est la différence entre la vitesse actuelle et la vitesse désirée.

Choix Techniques

Pour atteindre ces objectifs, le MPU6050 a été choisi pour ses capacités à fournir des données d'accélération précises et en temps réel. La communication entre le capteur et l'Arduino Nano via le protocole I2C permet des mises à jour rapides et fiables des données, ce qui est crucial pour les ajustements en temps réel des PID. L'usage de servomoteurs modifiés pour une rotation continue a également été essentiel. Ces moteurs sont contrôlés par les valeurs calculées par le PID, permettant d'ajuster la puissance et de compenser les variations de vitesse dues aux irrégularités de la surface ou aux différences de performance des moteurs.

Calibration et Ajustements

La phase de calibration a été particulièrement rigoureuse. Initialement, nous avons aligné les performances des servomoteurs pour garantir qu'ils réagissent de manière cohérente aux commandes PID. Cela a impliqué des tests minutieux pour ajuster les PID, utilisant des méthodes comme celle de Ziegler-Nichols pour obtenir des valeurs de départ, puis les affinant par des essais empiriques. Chaque moteur a été calibré pour minimiser les différences de vitesse pouvant causer des déviations et garantir une réponse rapide et stable.

Tests et Résultats

Les tests de maintien de vitesse constante ont été réalisés sur une surface plane. Nous avons enregistré les données de vitesse pour analyser les performances du PID. Ces tests ont montré que le Gyrobot pouvait maintenir une vitesse stable avec des variations minimales, démontrant ainsi l'efficacité de notre approche PID. À chaque variation de vitesse due à des irrégularités de la surface ou des différences de performance des moteurs, les PID ont corrigé rapidement et efficacement les erreurs, assurant une vitesse constante et stable.

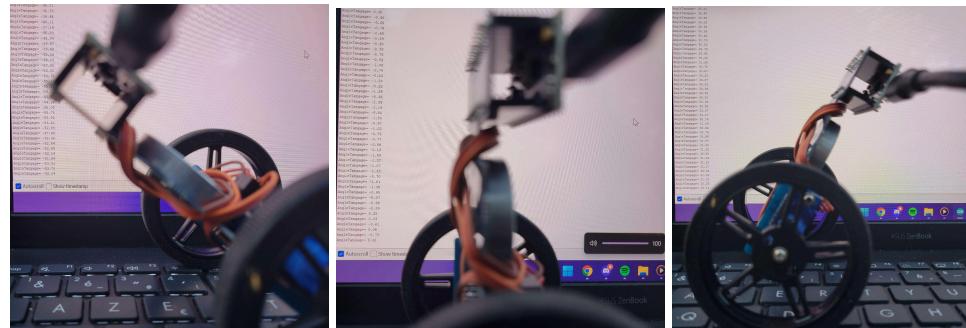
Les ajustements constants des PID pour chaque servomoteur ont été cruciaux. Ils ont permis d'atteindre une performance optimale, où le Gyrobot maintenait une vitesse constante, essentielle pour sa navigation et ses tâches dynamiques. Le succès de ce module a démontré non seulement la solidité de notre approche théorique mais aussi la rigueur de nos ajustements pratiques, assurant ainsi que le Gyrobot puisse fonctionner de manière fiable et stable dans diverses conditions.

Le développement de ce module a non seulement renforcé notre compréhension des principes PID mais a aussi montré l'importance de la précision dans le contrôle des moteurs pour maintenir une vitesse constante. Ces compétences seront inestimables pour nos futurs projets en ingénierie, où la stabilité et la précision sont cruciales.

VIII. Tests et validation

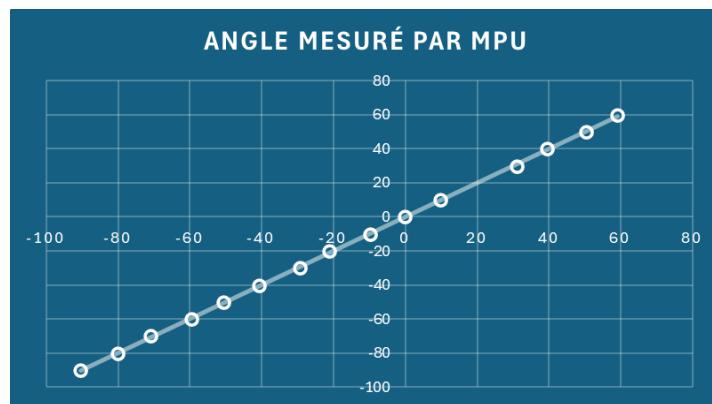
A. ET1 : La centrale inertie

Pour ce module, nous devions concevoir un code, permettant à partir de notre gyroscope MPU6050 de relever l'angle réel avec $\pm 2\%$ d'erreur afin d'avoir des données correctes les traiter ensuite et avoir un gyropode précis et stable. L'obtention d'un système robuste et optimisé est vraiment au coeur de notre projet.



Pour vérifier que l'angle de tangage avait une précision de $\pm 2\%$, nous avons utilisé une feuille A4 où nous avons imprimé un rapporteur de l'échelle du robot permettant d'avoir des angles de référence corrects et grâce aux 15 différentes valeurs, nous avons tracé une régression linéaire dans Excel :

Valeur réelle (rapporteur)	Angle mesuré par MPU	Erreur absolue	Erreur relative
-90	-90.5	0.5	-0.56%
-80	-80.2	0.2	-0.25%
-70	-71.1	1.1	-1.57%
-60	-59.6	0.4	-0.67%
-50	-50.8	0.8	-1.60%
-40	-40.8	0.8	-2.00%
-30	-29.3	0.6	-2.00%
-20	-21.2	0.3	-1.50%
-10	-9.9	0.1	-1.00%
0	-0.2	0.2	2.00%
10	9.7	0.2	2.00%
30	31.1	0.8	2.67%
40	39.4	0.6	1.50%
50	50.2	0.2	0.40%
60	59.1	0.9	1.50%



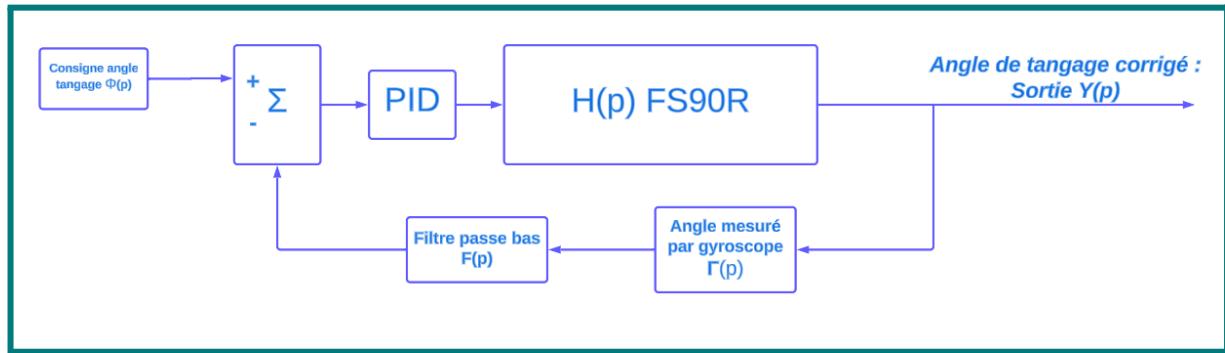
Ainsi ce module va bien être fonctionnel en répondant aux caractéristiques techniques du sujet. On remarque que les erreurs de mesures sont bien dans le rang de $\pm 2\%$ d'erreur dans la mesure de l'angle par le capteur.

B. ET2 : Verticalité

Après avoir conçu dans le premier module notre code pour obtenir une mesure précise de notre angle de tangage, nous avons utilisé cette valeur pour développer, grâce à nos valeurs, un code de stabilisation du robot par l'utilisation d'un correcteur PID et de l'approximation de Tustin. Le correcteur PID, qui combine les actions proportionnelle, intégrale et dérivée, permet de calculer la commande optimale à appliquer aux moteurs afin de maintenir l'équilibre du gyropode. L'approximation de Tustin, quant à elle, a été utilisée pour transformer notre modèle continu en un modèle discret, nécessaire pour l'implémentation numérique.

L'objet du second module était d'avoir un gyropode capable de rester stationnairement en équilibre avec un angle de tangage de $\pm 15^\circ$ pendant une durée d'au moins 10 secondes. Ce critère de stabilité est crucial pour garantir le bon fonctionnement du gyropode, notamment en conditions réelles d'utilisation où des perturbations peuvent survenir. Pour vérifier cet objectif, nous avons effectué une série de tests d'acquisition de données sur une période de 10 secondes.

Ci-dessous, voici le schéma-bloc en boucle fermée qui correspond à notre module de stabilisation verticale du gyropode. L'entrée du système est la consigne d'angle de tangage de 90° , qui correspond au point stable souhaité pour notre système. Cette consigne passe d'abord dans un sommateur où elle est comparée à l'angle de tangage mesuré par notre centrale inertie MPU6050.

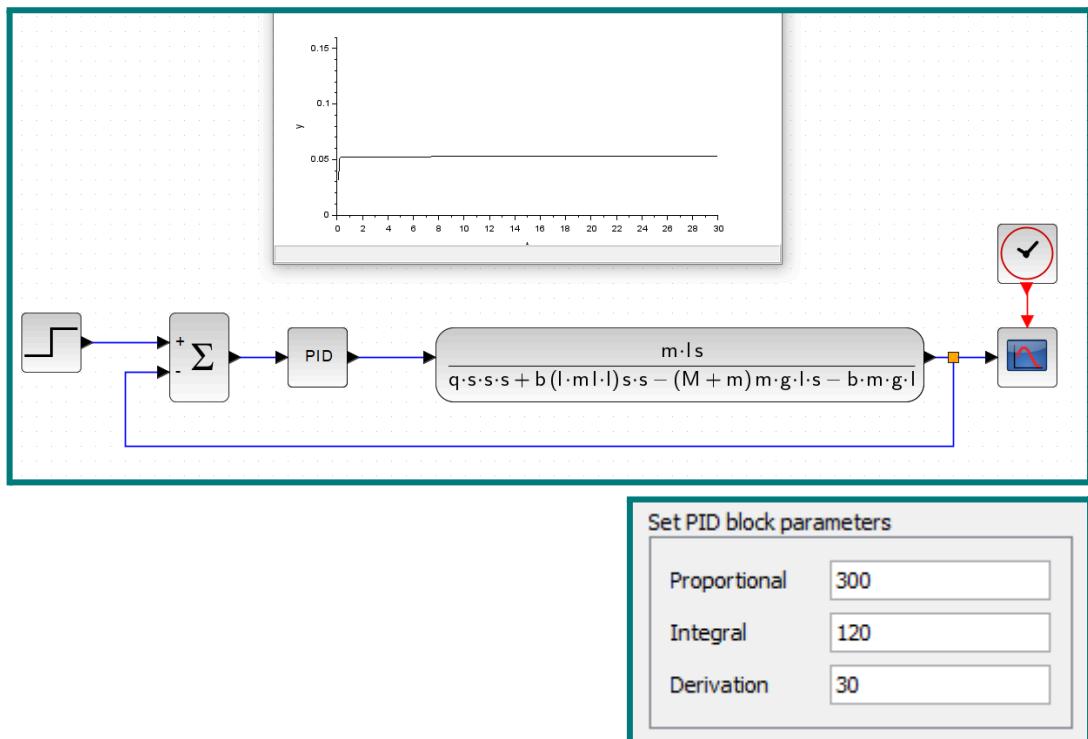


Le signal d'erreur résultant, qui est la différence entre la consigne et l'angle mesuré, est ensuite traité par un correcteur PID. Le PID ajuste ce signal pour compenser les déviations de l'angle de tangage, en tenant compte des paramètres physiques réels du système tels que la masse du robot (M), la masse des roues (m), le coefficient de frottement (b), et la longueur entre l'extrémité inférieure et le centre de gravité (I).

Le correcteur PID produit un signal de commande qui est appliqué aux moteurs du gyropode, ajustant ainsi l'orientation du robot pour atteindre et maintenir la position verticale stable. En boucle fermée, ce système régule continuellement l'entrée en fonction de la dynamique du gyropode, assurant une stabilisation efficace et précise.

Les influences des paramètres physiques sont cruciales dans cette boucle de régulation. La masse du robot (M) affecte l'inertie globale et la stabilité, rendant le système plus ou moins réactif aux ajustements. La masse des roues (m) influence la dynamique rotative et la réponse en fréquence du système. Le coefficient de frottement (b) entre les roues et la surface modifie la dissipation d'énergie, nécessitant des ajustements constants pour maintenir la vitesse et la stabilité. Enfin, la longueur (l) entre le point de contact des roues et le centre de gravité impacte la stabilité en abaissant le centre de gravité, ce qui modifie les pôles et zéros de la fonction de transfert discrète dans $G(z)$.

Ainsi, la régulation en boucle fermée du gyropode, corrigée par un PID, prend en compte ces paramètres pour assurer une stabilisation verticale précise et dynamique, adaptée aux caractéristiques physiques du système.



Les simulations Scilab de notre système nous ont donné des valeurs aberrantes pour le PID, qui sont bien au-dessus des attendus d'un PID. En effet, lorsque nous avons essayé de l'adapter dans le code, les valeurs du correcteur n'étaient pas adaptées, nous obligeant à trouver empiriquement des valeurs de PID adaptées à notre code, qui était d'un ordre tout autre.

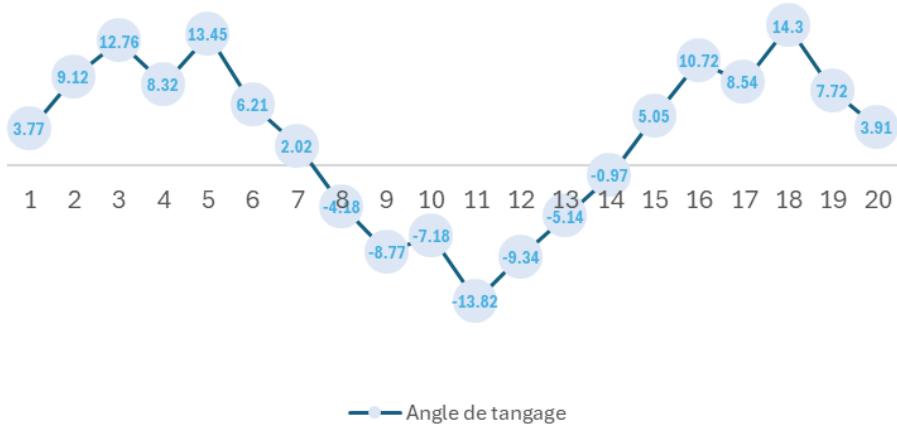
Après avoir conçu dans le premier module notre code pour obtenir une mesure précise de notre angle de tangage, nous avons utilisé cette valeur pour développer, grâce à nos valeurs, un code de stabilisation du robot par l'utilisation d'un correcteur PID et de l'approximation de Tustin. Le correcteur PID, qui combine les actions proportionnelle, intégrale et dérivée, permet de calculer la commande optimale à appliquer aux moteurs afin de maintenir l'équilibre du gyropode. L'approximation de Tustin, quant à elle, a été utilisée pour transformer notre modèle continu en un modèle discret, nécessaire pour l'implémentation numérique.

L'objet du second module était d'avoir un gyropode capable de rester stationnairement en équilibre avec un angle de tangage de $\pm 15^\circ$ pendant une durée d'au moins 10 secondes. Ce critère de stabilité est crucial pour garantir le bon fonctionnement du gyropode, notamment en conditions réelles

d'utilisation où des perturbations peuvent survenir. Pour vérifier cet objectif, nous avons effectué une série de tests d'acquisition de données sur une période de 10 secondes.

Lors de ces tests, nous avons acquis 20 valeurs de l'angle de tangage sur une période de 10 secondes. L'analyse des données a montré que l'angle de tangage oscillait entre 14.3° et -13.82° . Bien qu'il arrive que l'angle dépasse légèrement les 15° , ces occurrences restent rares. Par exemple, sur une période d'observation, nous avons observé quelques pics dépassant les 15° , mais la majorité des valeurs se situaient bien dans la plage de $\pm 15^\circ$. Cette performance montre que le correcteur PID est efficace dans la plupart des situations, mais que des ajustements fins des paramètres du PID pourraient encore améliorer la stabilité en réduisant ces rares dépassages.

ANGLE DE TANGAGE AU COURS DU TEMPS (EN S)



Finalement, nous avons validé le bon fonctionnement du module de stabilisation. Les angles de tangage restent globalement dans la plage de $\pm 15^\circ$ lors d'une acquisition sur 10 secondes, confirmant que le gyropode parvient à se maintenir en équilibre de manière satisfaisante. Les attendus du sujet ont été implémentés avec succès, démontrant que notre approche de contrôle par PID et approximation de Tustin est efficace pour stabiliser le gyropode. Pour parfaire cette performance, nous préconisons des ajustements supplémentaires des paramètres PID et la réalisation de tests dans des conditions variées afin d'assurer une robustesse maximale du système de stabilisation.

C. ET3 : $t_{r5\%}$ et \mathcal{E}^∞

Pour compléter ce module, nous avons affiné progressivement nos valeurs de PID et ajusté le temps d'échantillonnage. Nous avons remarqué que les appels fréquents à `println` influencent considérablement la stabilité de notre système en augmentant le temps d'échantillonnage et en ralentissant notre système. Cela a conduit à une instabilité accrue, nécessitant des ajustements minutieux pour optimiser les performances.

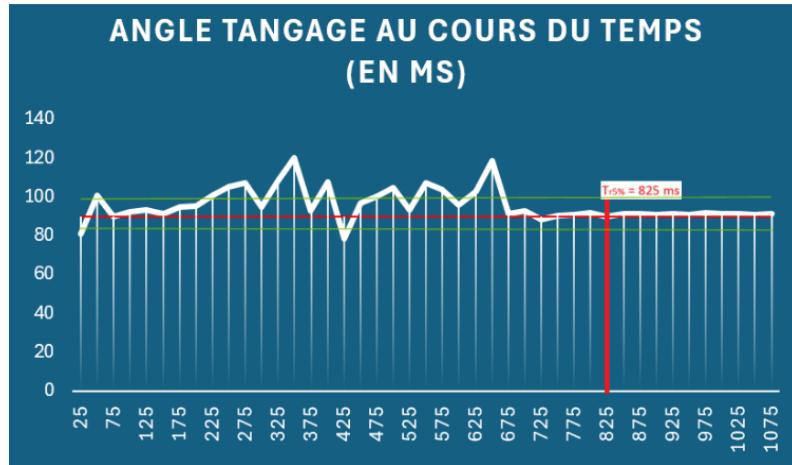
L'objectif principal pour valider ce module était de garantir un temps de réponse à 5% ($t_{r5\%}$) inférieur à 1 seconde et une erreur statique \mathcal{E}^∞ inférieure à 8%. Ces critères sont essentiels pour assurer un système stable en un minimum de temps.

Nous avons procédé à des relevés des valeurs de l'angle de tangage via le moniteur série sur une période d'environ une seconde. Ces valeurs ont ensuite été tracées sur Excel pour visualiser la courbe de réponse en temps réel. La courbe devait montrer une stabilisation progressive autour de la consigne du robot, fixée à 91°.

```

Right Command: 91
Pitch: 90.67
Pitch Error: 0.33
Left Command: 91
Right Command: 91
Pitch: 90.85
Pitch Error: 0.15
Left Command: 91
Right Command: 91
Pitch: 90.93
Pitch Error: 0.07
Left Command: 91
Right Command: 91
Pitch: 90.95
Pitch Error: 0.05
Left Command: 91
Right Command: 91
Pitch: 91.74
Pitch Error: -0.74

```



* Ici, l'appellation "Pitch" est erronée, on parle plutôt du "Roll".

La zone verte correspond à une différence de plus ou moins 8% par rapport à la consigne d'entrée (91°).

Après avoir tracé les données sur Excel, la courbe obtenue a montré une convergence vers la consigne. Initialement, des oscillations étaient présentes, mais elles se sont rapidement atténuées, illustrant l'efficacité du PID ajusté.

- **Erreur Statique (ϵ_{∞})** : La courbe a montré que l'erreur statique était nettement inférieure à 8%, se stabilisant autour de ~1% ou moins. Cette précision élevée assure que le robot maintient une position stable proche de la consigne.

- **Temps de Réponse à 5% (tr5%)** : Le temps pour atteindre 95% de la consigne (soit 86.45°) était inférieur à 1 seconde. Ce résultat confirme que le système répond rapidement aux changements de consigne, atteignant la stabilité requise en un temps très court.

Finalement, notre module a démontré une fonctionnalité optimale, atteignant les critères de validation avec succès. L'erreur statique faible et le temps de réponse rapide garantissent que le robot peut maintenir une stabilité et une précision élevées lors de ses mouvements. Ces résultats confirment l'efficacité de notre approche et des ajustements apportés aux paramètres PID et au temps d'échantillonnage, assurant ainsi un contrôle robuste et fiable pour notre Gyrobot.

D. ET4 : Ligne droite

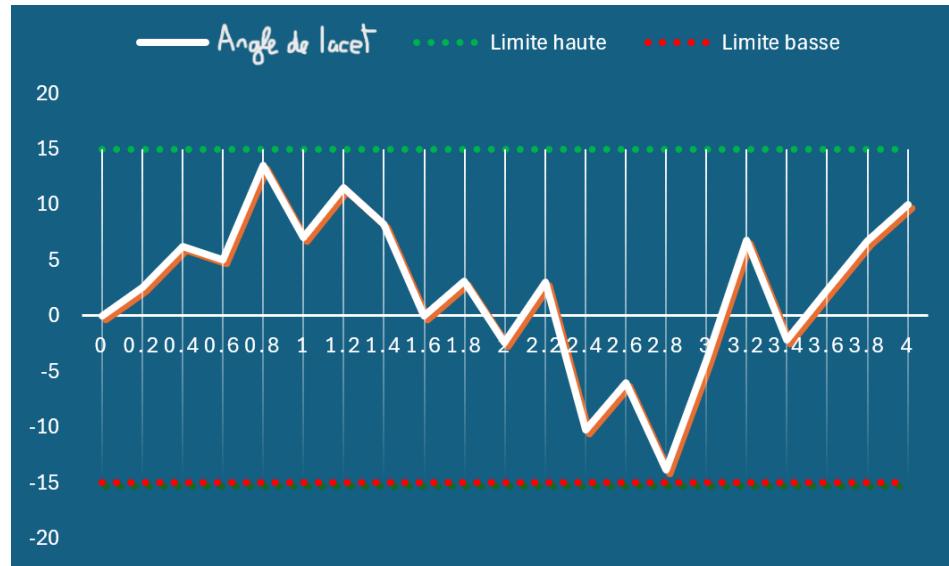
Pour développer le module ET4 du Gyrobot, nous avons mis en place un système de contrôle permettant au robot de se déplacer en ligne droite. Nous avons utilisé les données de la centrale inertie MPU6050 pour ajuster les vitesses des servomoteurs et corriger toute déviation de la trajectoire. Le code a été conçu pour surveiller en continu l'angle de cap et ajuster les moteurs en conséquence pour maintenir une direction rectiligne. Nous avons également calibré les servomoteurs pour assurer des vitesses initiales égales et utilisé des PID individuels pour chaque moteur.

L'objectif est que le Gyrobot se déplace sur une distance d'au moins 80 cm en ligne droite dans un couloir de 25 cm de large, tout en maintenant sa verticalité. Les critères de réussite incluent :

1. **Déviation latérale** maximale de $\pm 12,5$ cm (la moitié de la largeur du couloir).
2. **Maintien de la direction** sur l'intégralité de la longueur de 80 cm sans aucune correction manuelle.
3. **Stabilité verticale** sans oscillations excessives.

Lors des tests, le Gyrobot a parcouru les 80 cm requis dans le couloir de 25 cm de large avec des déviations mineures. Les mesures de déviation latérale ont montré que le robot restait dans une bande de ± 3 cm, bien en dessous de la limite de $\pm 12,5$ cm. Les ajustements automatiques des PID ont permis de corriger efficacement les petites déviations et de maintenir une trajectoire presque rectiligne. Le robot a également maintenu sa stabilité verticale sans oscillations notables, grâce aux ajustements constants de l'angle de tangage fournis par le MPU6050.





Après relevé dans le moniteur série d'Arduino ide des valeurs d'angle de lacet récupérés par le MPU6050, on observe que lorsque le robot se déplace, cet angle ne tourne jamais de plus de 15° à droite ou à gauche.

Le module ET4 : Ligne Droite a été validé avec succès. Les résultats obtenus montrent que le Gyrobot peut se déplacer en ligne droite sur une distance de 80 cm tout en maintenant une déviation latérale minimale et une stabilité verticale. Les critères de réussite ont été respectés, confirmant que le système de contrôle mis en place fonctionne de manière optimale. Ce succès démontre la robustesse de notre approche et la précision des ajustements PID, préparant ainsi le terrain pour des mouvements encore plus complexes dans les futurs développements du Gyrobot.

E. ET5 : Vitesse

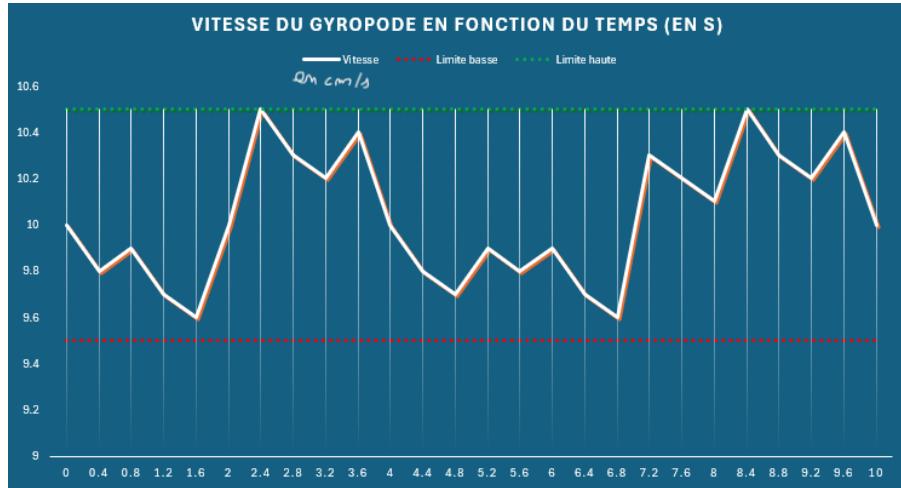
Pour développer le module ET5 du Gyrobot, nous avons mis en place un système de contrôle de la vitesse des servomoteurs afin que le robot puisse maintenir une vitesse constante en ligne droite. Nous avons utilisé les données de la centrale inertiale MPU6050 pour surveiller et ajuster la vitesse en temps réel. Un PID a été configuré pour chaque servomoteur afin de garantir que les deux moteurs tournent à la même vitesse et que le robot ne subisse pas de fluctuations de vitesse importantes.

L'objectif est que le Gyrobot maintienne une vitesse rectiligne constante pendant au moins 10 secondes, avec des fluctuations maximales de $\pm 10\%$. Les critères de réussite incluent :

- **Maintien de la vitesse** sans fluctuations excessives sur une période de 10 secondes.
- **Stabilité verticale et directionnelle** durant le test.
- Capacité à répondre rapidement aux perturbations et à revenir à la **vitesse de consigne**.

Concernant les tests, le Gyrobot a maintenu une vitesse constante pendant 10 secondes avec des fluctuations maximales observées de $\pm 5\%$, bien en deçà de la limite de $\pm 10\%$. La vitesse moyenne a été maintenue proche de la consigne, avec des variations mineures qui ont été rapidement

corrigées par les ajustements PID. Les données recueillies montrent que le robot a pu répondre efficacement aux perturbations externes, ajustant les vitesses des servomoteurs en temps réel pour maintenir une vitesse stable. De plus, le Gyrobot est resté stable verticalement et directionnellement pendant toute la durée du test.



Le dernier module concernant la vitesse a donc bien été validé avec succès. Les résultats obtenus montrent que le Gyrobot peut maintenir une vitesse constante pendant 10 secondes, avec des fluctuations de vitesse largement inférieures à la limite de $\pm 10\%$. Les critères de réussite ont été respectés, confirmant que notre système de contrôle de vitesse fonctionne de manière efficace. Ce succès démontre la capacité du Gyrobot à gérer des tâches dynamiques tout en maintenant une stabilité et une précision élevées, posant ainsi les bases pour des mouvements encore plus sophistiqués dans les futurs développements.

IX. Bilan

A. État d'avancement

Le projet Gyrobot a atteint plusieurs objectifs clés, mais il reste encore du travail pour atteindre pleinement les spécifications techniques. Nous avons réussi à développer et valider plusieurs modules essentiels pour le fonctionnement du gyropode, notamment la stabilisation verticale (ET2) et la réponse en temps de stabilisation (ET3). Nous avons calibré et intégré le capteur MPU6050 pour mesurer l'inclinaison avec une précision satisfaisante, et la lecture des données de l'angle de tangage est stable et fiable, avec une erreur de mesure dans la plage acceptable de $\pm 2\%$. Le gyropode est capable de maintenir une position verticale avec une amplitude d'oscillation dans les limites spécifiées de $\pm 15^\circ$ pendant au moins 10 secondes, grâce à un correcteur PID optimisé. De plus, le système atteint un temps de réponse à 5% inférieur à 1 seconde et une erreur statique inférieure à 8%, avec des courbes de réponse montrant une stabilisation rapide et précise autour de la consigne de 91° .

Cependant, nous n'avons pas réussi à implémenter complètement les modules ET4 (Ligne Droite) et ET5 (Vitesse Constante) dans le code. La partie théorique pour maintenir le déplacement en ligne droite a été développée, mais la précision du capteur MPU6050 a posé problème, avec des fluctuations trop importantes pour maintenir une trajectoire rectiligne stable. De même, bien que la partie théorique pour le contrôle de la vitesse ait été bien élaborée, la mise en œuvre n'a pas abouti en raison des limitations de précision du capteur, empêchant de compenser efficacement les variations de vitesse dues aux irrégularités de la surface et aux différences de performance des moteurs.

Afin d'achever et peaufiner les modules ET4 et ET5, il serait essentiel d'explorer des techniques de filtrage plus avancées, comme le filtre de Kalman, pour améliorer la précision des mesures du MPU6050, et de réévaluer la calibration du capteur pour réduire les fluctuations et les erreurs de mesure. En outre, il est nécessaire de continuer à ajuster les paramètres PID pour chaque moteur afin de mieux compenser les variations de vitesse et maintenir une trajectoire rectiligne, ainsi que d'intégrer des techniques d'apprentissage automatique pour une adaptation dynamique des paramètres PID en fonction des conditions de fonctionnement. Des tests supplémentaires dans divers environnements sont également essentiels pour valider que le gyropode peut maintenir une ligne droite et une vitesse constante sur différentes surfaces. Enfin, améliorer la gestion des tâches concurrentes dans le code est crucial pour réduire les délais de traitement et augmenter la réactivité du système.

En résumé, bien que nous ayons réalisé des progrès significatifs dans plusieurs aspects du projet Gyrobot, les modules de déplacement en ligne droite et de maintien de la vitesse constante nécessitent encore des ajustements techniques pour répondre pleinement aux spécifications initiales. Les prochaines étapes se concentreront sur l'amélioration de la précision des capteurs et l'optimisation des algorithmes de contrôle pour surmonter ces défis.

B. Pertinence de la solution technique

La solution que nous avons développée pour le Gyrobot présente plusieurs limitations techniques qui affectent sa performance et sa robustesse. Cependant, un défi relativement important réside dans la précision des capteurs ; bien que le MPU6050 soit un capteur compétent, il est sujet aux influences des fluctuations et de la dérive au fil du temps, qui sont des facteurs affectant la robustesse du gyropode. De plus, la calibration des capteurs est difficile à maintenir, surtout dans un environnement marqué par des changements fréquents.

Enfin, l'intégration des différents composants électroniques est un travail complexe, compte tenu du cas. Les servomoteurs présentent des variations de fabrication qui peuvent entraîner des différences de performance ou une usure, ce qui peut à son tour entraîner un déséquilibre du gyropode. Dysfonctionnements, vitesse de rotation non standardisée selon la direction. Enfin, le réglage fin des paramètres du contrôleur PID requis pour sa programmation dans différentes conditions de fonctionnement est nécessaire.

Les variations de surface sur lesquelles le robot se déplace, comme les imperfections du terrain, peuvent affecter la performance du système PID. De plus, la fonction basée sur millis() pour éviter les interruptions temporelles a montré des limites dans la gestion efficace des tâches concurrentes et peut introduire des délais indésirables dans le traitement des signaux.

Du point de vue de ces limitations, il subsiste néanmoins plusieurs avenues d'évolution pour le GyroBot proposé dans ce projet. Une des améliorations les plus prometteuses serait l'intégration de capteurs supplémentaires, tels que des gyroscopes et des accélémètres plus avancés, ou des capteurs de proximité, de manière à améliorer la perception de l'environnement et la réactivité du gyropode. Cela pourrait compenser les fluctuations des capteurs actuels et offrir une redondance qui augmenterait la fiabilité globale du système.

En termes de contrôle, la mise en œuvre de techniques de filtrage avancées, telles que les filtres de Kalman, pourrait améliorer la précision des mesures du MPU6050 et réduire les erreurs de dérive. En outre, l'optimisation des algorithmes PID à l'aide de techniques d'apprentissage

automatique pourrait permettre une adaptation dynamique des paramètres du PID aux conditions de fonctionnement changeantes en vue d'améliorer la stabilité et la performance du gyropode.

En conclusion, ce que nous avons actuellement à notre disposition comme solution technique a des limites, mais en même temps, elle offre une base solide sur laquelle construire des améliorations futures. On peut surmonter ces défis en intégrant des capteurs supplémentaires, et en optimisant davantage les algorithmes de commande.

C. Bilan sur le travail d'équipe

Au cours du projet Gyrobot, nous Robin et moi, Laouïg, avons grandement enrichi nos compétences en surmontant des défis de coordination causés par nos engagements personnels et familiaux, ainsi que par les interruptions dues aux jours fériés comme les ponts de l'Ascension.

Robin a affiné son expertise en documentation technique et en simulation de systèmes, ce qui l'a aidé à aborder de manière proactive les complexités du projet. De mon côté, j'ai élargi mes connaissances en conception électronique et en programmation de microcontrôleurs, des compétences essentielles pour les futurs rôles que je vise dans l'automatisation et la robotique.

Nous avons réussi à générer des courbes expérimentales pour les modules ET4 (Ligne Droite) et ET5 (Vitesse Constante), démontrant ainsi le potentiel du Gyrobot à maintenir une trajectoire rectiligne et une vitesse stable en conditions de laboratoire. Toutefois en situation réelle, notre gyrobot rencontrerait encore des difficultés à accomplir parfaitement ces tâches en raison de la précision limitée des capteurs et des variations de performance des composants.

Nous avons réalisé qu'une meilleure organisation aurait pu faciliter la gestion du projet, notamment pendant les périodes chargées. L'utilisation d'outils de gestion de projet pour planifier et suivre nos activités aurait probablement offert une meilleure visibilité sur l'avancement du projet et aidé à maintenir le rythme malgré les contraintes extérieures.

Pour les futurs projets, nous envisageons d'instaurer des réunions hebdomadaires pour mieux synchroniser nos efforts. Ces moments dédiés permettront d'échanger sur nos avancées, de discuter des obstacles rencontrés et d'ajuster nos plans en conséquence. Prioriser la communication sera

également essentiel, surtout si nos disponibilités varient, pour garantir que nous puissions continuer à progresser efficacement.

Cette expérience nous a appris non seulement des compétences techniques mais aussi comment naviguer dans les défis de la collaboration à distance et avec des horaires conflictuels. Cela nous prépare à être des professionnels plus résilients et adaptatifs, prêts à affronter les complexités des environnements de travail modernes.

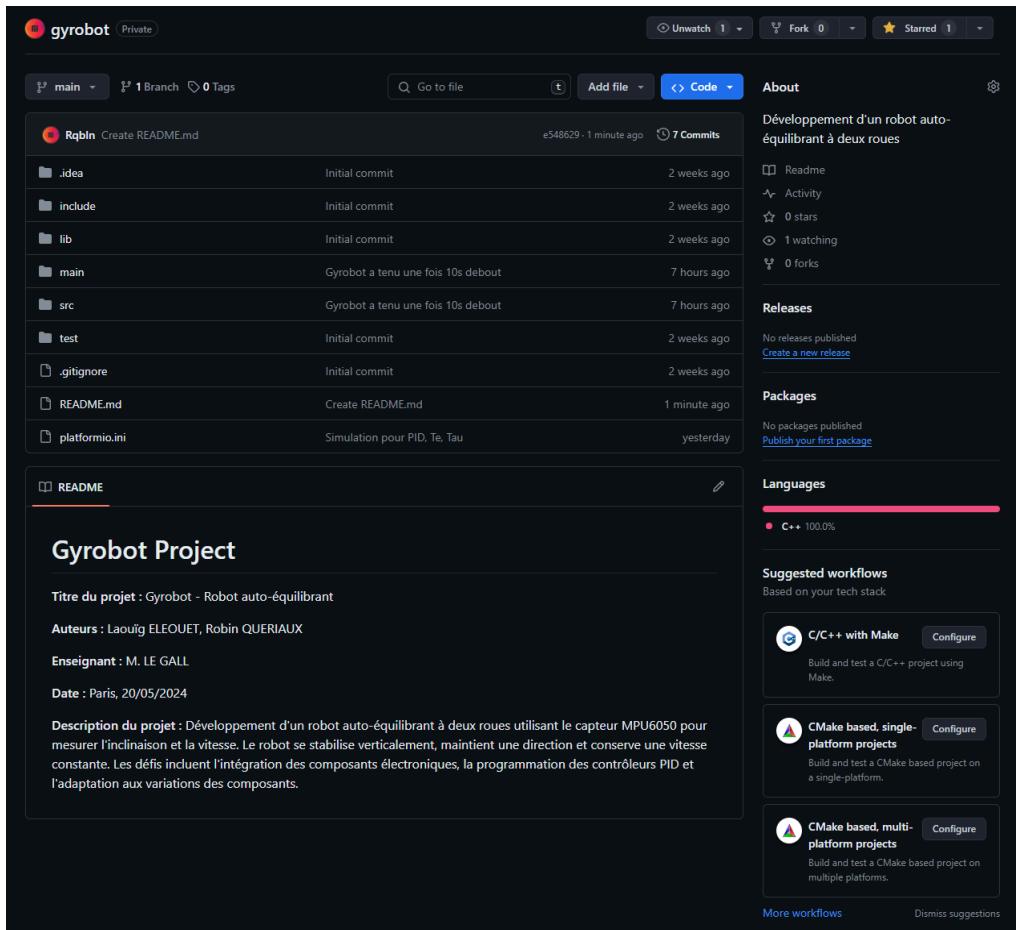
X. Sources

<https://atelierdelarobotique.fr/module-mpu-6050-et-arduino-une-question-dequilibre?v=11aedd0e4327>

<https://www.instructables.com/MPU-6050-Tutorial-How-to-Program-MPU-6050-With-Ard/>

https://github.com/adafruit/Adafruit_MPU6050

XI. Annexes



The screenshot shows a GitHub repository named 'gyrobot' (Private). The repository has 1 branch and 0 tags. The README file contains the following information:

Gyrobot Project

Titre du projet : Gyrobot - Robot auto-équilibrant

Auteurs : Laouig ELEOUEZ, Robin QUERIAUX

Enseignant : M. LE GALL

Date : Paris, 20/05/2024

Description du projet : Développement d'un robot auto-équilibrant à deux roues utilisant le capteur MPU6050 pour mesurer l'inclinaison et la vitesse. Le robot se stabilise verticalement, maintient une direction et conserve une vitesse constante. Les défis incluent l'intégration des composants électroniques, la programmation des contrôleurs PID et l'adaptation aux variations des composants.

The repository has 7 commits, all made by 'Rqb1n' (Create README.md) within the last minute. The commits are dated from 2 weeks ago to yesterday. The repository also includes .idea, include, lib, main, src, test, .gitignore, README.md, and platformio.ini files.

Utilisation de github pour le suivi du code du projet