

Presentation du Projet Metier

Mise en oeuvre et stabilisation d'un bras de drone quadrirotor

Ahmed Amine NOUABI

Mamane Bello Abdoul Hakim

Encadre par: Mr. TALEB

Jury: Mr. TALEB Mr.SAADI Mr. LAGRIOUI

1st July 2024



المدرسة الوطنية العليا للفنون و المهن
ÉCOLE NATIONALE SUPÉRIEURE D'ARTS ET MÉTIERS



Table of Contents

1 Introduction

- ▶ Introduction
- ▶ Conception du systeme
- ▶ Conception Logiciel
- ▶ Conclusion



Objectifs du projet.

1 Introduction

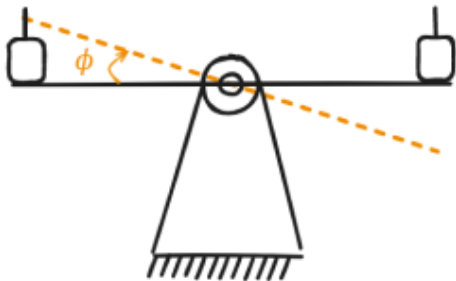
- Stabiliser un bras rotative.
- Utilisation d'un capteur MPU6050 et un algorithme de calcul d'angle.
- Mise en oeuvre d'un contrôleur PID pour commander les moteurs.



Presentation du projet.

1 Introduction

Le projet consiste en la conception et la mise en œuvre d'un système de stabilisation pour une barre rotative à un degré de liberté. En utilisant un microcontrôleur Arduino et un capteur IMU MPU6050, nous avons développé un algorithme de contrôle PID pour maintenir la barre stable malgré les perturbations externes.



Fonctionnement du systeme.

1 Introduction

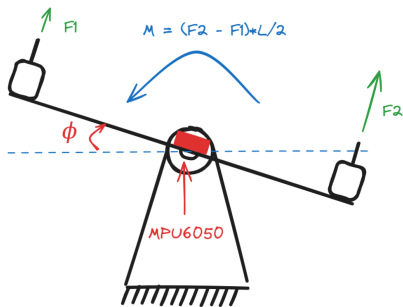


Figure: Fonctionnement du systeme.



Table of Contents

2 Conception du systeme

- ▶ Introduction
- ▶ Conception du systeme
- ▶ Conception Logiciel
- ▶ Conclusion

Conception mécanique.

2 Conception du système

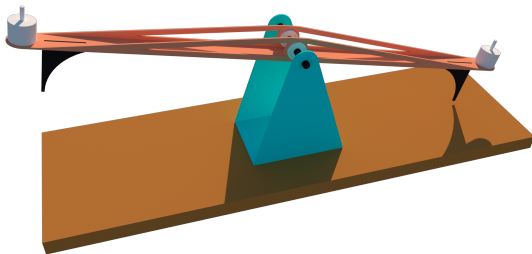
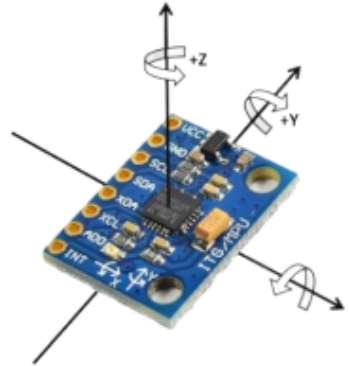


Figure: Modele mecanique 3D.

Calcul d'angle

2 Conception du systeme

- Le MPU6050 est un capteur qui combine un accéléromètre et un gyroscope.
- L'accéléromètre mesure l'accélération linéaire tandis que le gyroscope mesure la vitesse angulaire.





Calcul d'angle par gyroscope

2 Conception du système

Modele mathematique du gyroscope

$$\omega = \frac{d\phi}{dt}$$



Calcul d'angle par gyroscope

2 Conception du système

Modele mathematique du gyroscope

$$\omega = \frac{d\phi}{dt}$$

Calcul d'angle par gyroscope

$$\hat{\phi}_{gyro}(n) = \sum_{i=0}^n \omega(i) * T$$

- T est le temps d'échantillonnage $0.004s$. ($f_e = 250Hz$)



Limitation du modele

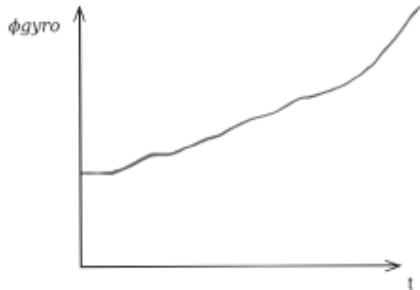
2 Conception du systeme

- Le gyroscope est sujet à des dérives.
- L'angle calculé par le gyroscope diverge avec le temps.

Modele proche de la realite

$$\omega = \frac{d\phi}{dt} + b_g(t) + n_g(t)$$

- $b_g(t)$ est l'offset du gyroscope.
- $n_g(t)$ est le bruit du gyroscope.





Calcul d'angle par accéléromètre

2 Conception du système

Modele mathématique de l'accéléromètre

$$\vec{a} = -R * \vec{g}$$

Calcul d'angle par accéléromètre

$$a_x = g \sin(\phi)$$

$$a_y = -g \sin(\theta) \cos(\phi)$$

$$a_z = -g \cos(\theta) \cos(\phi)$$

$$\hat{\phi}_{accelo}(n) = \arcsin\left(\frac{a_x(n)}{g}\right)$$



Fusion des resultats

2 Conception du systeme

- Pour pallier les dérives du gyroscope, on combine les résultats de l'accéléromètre et du gyroscope.
- En prenant α tres proche de 1, on donne plus de poids au gyroscope. ($\alpha = 0.9996$)

Fusion des resultats

$$\hat{\phi}(n) = \alpha * \hat{\phi}_{gyro}(n) + (1 - \alpha) * \hat{\phi}_{acelo}(n)$$



Commode des moteurs

2 Conception du système

- on utilise un ESC (Electronic Speed Controller) pour commander le moteur.
- ESC est commandé par un signal PWM (Pulse Width Modulation).

Plage de commande

$$1000\mu s \leq T_{on} \leq 2000\mu s$$



Contrôleur PID

2 Conception du système

Equation du contrôleur PID

$$u(t) = K_p * e(t) + K_i * \int e(t)dt + K_d * \frac{de(t)}{dt}$$

- Pour utiliser une commande symétrique. $\gamma = 1500 \pm u(t)$
- γ valeur de la commande PWM du moteur.
- On limite $u(t)$ entre -400 et 400 .

Schema de controle

2 Conception du systeme

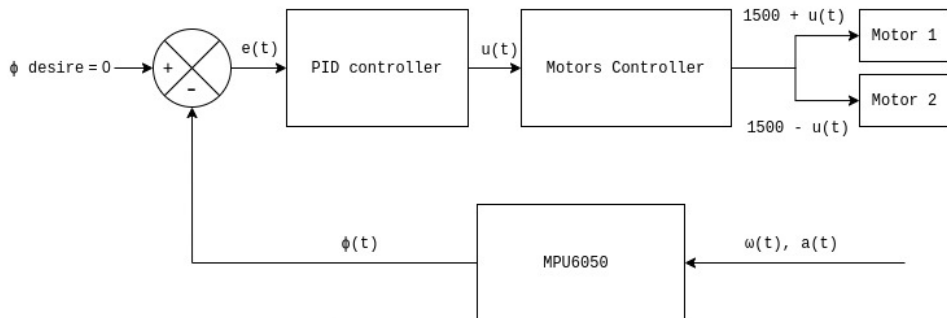


Figure: Schema de controle.

Montage Electrique

2 Conception du systeme

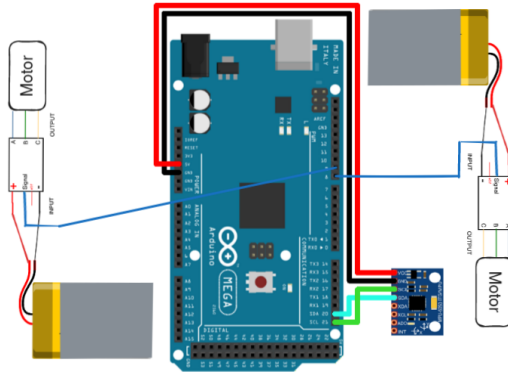


Figure: Montage Electrique.



Table of Contents

3 Conception Logiciel

- ▶ Introduction
- ▶ Conception du systeme
- ▶ **Conception Logiciel**
- ▶ Conclusion



Conception Logiciel

3 Conception Logiciel

- Pour faciliter la maintenance et l'évolution du code, on a utilisé une approche orientée objet (OOP).
- On utilise la librairie `Wire` pour la communication I2C avec le capteur MPU6050.
- On utilise la librairie `ESC` pour la commande PWM des moteurs.

Composants Logiciel OOP

3 Conception Logiciel

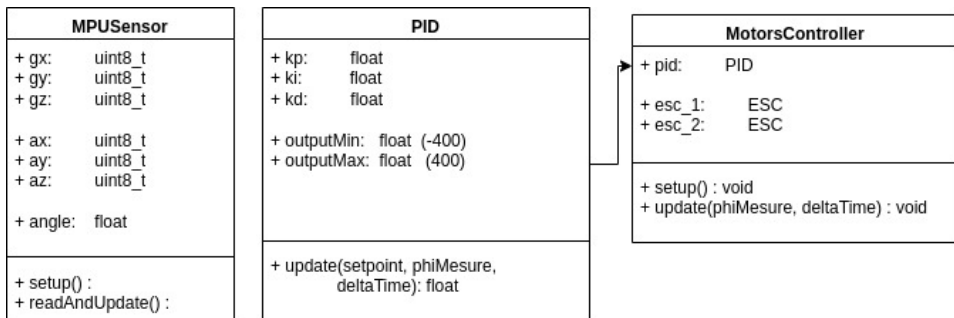


Figure: UML des Composants Logiciel.



Code Arduino - Setup

3 Conception Logiciel

```
MPUSensor mpu;  
MotorsController motors;  
  
double deltaTime_micros = 0.004 / 1000000.0;  
  
void setup() {  
    mpu.setup();  
    motors.setup();  
}
```



Code Arduino - Loop

3 Conception Logiciel

```
void loop() {  
    current_micros = micros();  
  
    mpu.readAndUpdate();  
    motors.update(mpu.angle, deltaTime);  
  
    // Pour assurer une boucle de controle de 4ms.  
    while (micros() - current_micros < deltaTime_micros);  
}
```



Table of Contents

4 Conclusion

- ▶ Introduction
- ▶ Conception du systeme
- ▶ Conception Logiciel
- ▶ Conclusion



Conclusion

4 Conclusion

- En partant d'ici on peut envisager l'implementation ou de creer un environnement de simulation pour tester le systeme.
- On peut aussi considerer a ajouter differentes strategies de controle et d'acquisition pour un benchmark de resultats.



Merci pour votre attention!