Documentation Complète : Calcul de la Vitesse et de la Position d'un Robot à Deux Roues

Introduction

Cette documentation détaille de manière exhaustive la méthode utilisée pour calculer la vitesse, la position et l'orientation d'un robot mobile à deux roues motrices, en s'appuyant sur le code de simulation développé.

1. Modèle Théorique d'un Robot à Deux Roues

Un robot différentiel se déplace grâce à deux roues indépendantes situées de part et d'autre de son châssis. Chaque roue est actionnée par un moteur, et la combinaison des vitesses des deux roues détermine la trajectoire du robot.

Paramètres Clés :

- RR : Rayon de la roue.
- LL : Distance entre les roues.
- vgv_g : Vitesse de la roue gauche.
- vdv_d : Vitesse de la roue droite.
- $\theta \mid theta$: Orientation du robot.

Relations Fondamentales :

- La vitesse linéaire vv est la moyenne des vitesses des deux roues.
- La vitesse angulaire $\omega \mid omega$ est la différence des vitesses divisée par l'écartement des roues.

2. Analyse du Code de Simulation

Le code commence par récupérer les vitesses des moteurs et les convertit en vitesses linéaires :

```
left_velocity = (left_speed / 360.0) * (2 * math.pi *
self.WHEEL_RADIUS)
right_velocity = (right_speed / 360.0) * (2 * math.pi *
self.WHEEL_RADIUS)
```

Explication :

- La vitesse en tours par minute (RPM) est transformée en radians par seconde, puis multipliée par la circonférence de la roue.

Le code calcule ensuite :

- **Vitesse linéaire :** $v=(vg+vd)/2v$
- **Vitesse angulaire :** $\omega=(vd-vg)/L$

3. Cas de Mouvement Traités

- **Arrêt complet :** Les deux vitesses sont nulles.
- **Rotation sur place :** Vitesse de la roue gauche = - Vitesse de la roue droite.
- **Pivot sur une roue :** Une roue fixe, l'autre active.
- **Déplacement linéaire :** Les deux roues à la même vitesse.
- **Mouvement en courbe :** Vitesses différentes mais non opposées.

4. Calcul de la Position et de l'Orientation

Le code met à jour les coordonnées et l'angle :

- $x=x+v\times\cos(\theta)\times dt$
- $y=y+v\times\sin(\theta)\times dt$
- $\theta=\theta+\omega\times dt$

5. Défis Rencontrés et Solutions

- **Synchronisation temporelle :** Utilisation d'une horloge basée sur `time.time()`.
- **Gestion du pivot :** Ajout d'une condition spécifique dans le code.
- **Vérification des collisions et limites :** Prévention des dépassements dans l'environnement simulé.

7. Conclusion

Cette documentation offre une vue complète sur le fonctionnement d'un robot à deux roues et la logique de calcul de sa trajectoire. Elle couvre les aspects théoriques, l'analyse du code et les améliorations potentielles.