



Mauro José Silva Sousa

Cinemática com CoppeliaSim

Campina Grande
25 de junho de 2024

Sumário

INTRODUÇÃO	1
1 Controle Cinemático de Robôs Móveis	2
2 Cinemática Direta e Cinemática Inversa	3
2.1 Cinemática Direta	3
2.2 Cinemática Inversa	4
3 Modelagem Cinemática de Robôs Holonômicos e Não-Holonômicos	5
3.1 Robôs Holonômicos	5
3.2 Robôs Não-Holonômicos	5
4 Modelo Cinemático de Robôs de Tração Diferencial	6
5 Implementação das Equações Cinemáticas	7
5.1 Equações Utilizadas	7
5.2 Implementação em Python	7
5.2.1 Configuração da API Remota	7
5.2.2 Controle das Velocidades das Rodas	8
5.2.3 Controle do Robô na Simulação	8
CONCLUSÃO	10

INTRODUÇÃO

O controle cinemático é fundamental na robótica móvel, permitindo o movimento preciso de robôs. Este relatório aborda o controle cinemático de um robô de tração diferencial usando CoppeliaSim. Exploramos conceitos como cinemática direta e inversa, modelagem de robôs holonômicos e não-holonômicos, e implementação das equações cinemáticas. O objetivo é compreender como o controle cinemático afeta o movimento do robô. Apresentamos uma visão geral desses conceitos e sua aplicação prática, visando auxiliar estudantes e profissionais interessados em robótica móvel.

1 Controle Cinemático de Robôs Móveis

O controle cinemático é essencial para o movimento preciso e eficiente de robôs móveis. Ele determina como as velocidades das rodas ou juntas do robô se traduzem em movimento no ambiente, garantindo que o robô possa se locomover de forma autônoma e segura. Este relatório aborda o controle cinemático de um robô de tração diferencial usando CoppeliaSim, visando compreender como esse controle afeta o movimento do robô.

2 Cinemática Direta e Cinemática Inversa

A cinemática direta e inversa são conceitos fundamentais na robótica móvel, essenciais para o controle de movimento de robôs. Aqui está uma introdução para a seção sobre esses tópicos:

2.1 Cinemática Direta

A cinemática direta é o processo de determinar a posição e orientação do robô com base nas velocidades das rodas ou juntas. Para um robô de tração diferencial, as equações cinemáticas diretas são:

$$v = \frac{V_r + V_l}{2}$$

$$\omega = \frac{V_r - V_l}{L}$$

Onde:

- v é a velocidade linear do robô, escrito da forma:

$$v = \omega \cdot r$$

- ω é a velocidade angular do robô;
- V_r é a velocidade da roda direita, escrita da forma:

$$V_r = \omega_r \cdot \left(r + \frac{L}{2}\right) \quad (1)$$

- V_l é a velocidade da roda esquerda, escrita da forma:

$$V_r = \omega_l \cdot \left(r - \frac{L}{2}\right) \quad (2)$$

- r é o raio da roda
- L é a distancia entre as rodas.

2.2 Cinemática Inversa

A cinemática inversa é o processo de determinar as velocidades das rodas ou juntas necessárias para alcançar uma determinada posição e orientação. Para um robô de tração diferencial, as equações cinemáticas inversas são:

$$\omega_r = \frac{2v + \omega L}{2r}$$

$$\omega_l = \frac{2v - \omega L}{2r}$$

Essas equações são essenciais para o controle preciso do movimento do robô, permitindo que ele se mova de forma suave e eficiente em seu ambiente.

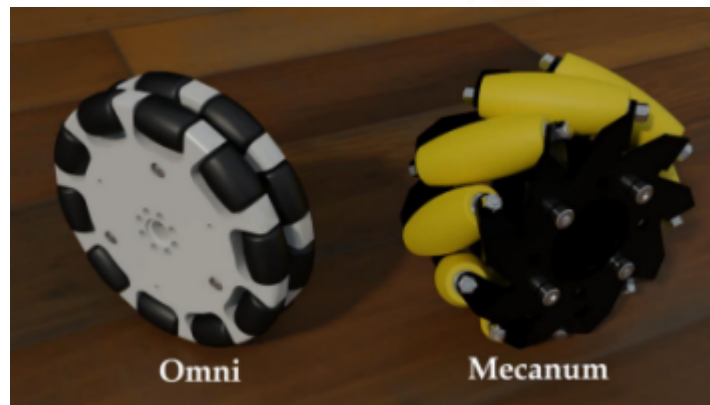
3 Modelagem Cinemática de Robôs Holonômicos e Não-Holonômicos

Os robôs móveis podem ser classificados em duas categorias principais: holonômicos e não-holonômicos. Os robôs holonômicos são capazes de se mover livremente em todas as direções, enquanto os não-holonômicos possuem restrições em seu movimento.

3.1 Robôs Holonômicos

Os robôs holonômicos podem se mover em qualquer direção sem restrições. Para esses robôs, a cinemática direta e inversa é mais simples, envolvendo o cálculo das velocidades das rodas ou juntas necessárias para alcançar uma determinada velocidade e direção do robô. Exemplos de robôs holonômicos incluem robôs com rodas omnidirecionais e Mecanum.

Figura 1: Rodas holonômicas



3.2 Robôs Não-Holonômicos

Os robôs não-holonômicos possuem restrições em seu movimento, o que significa que não podem se mover livremente em todas as direções. Um exemplo comum de robô não-holonômico é o robô de tração diferencial, que pode se mover para frente, para trás e girar em torno do próprio eixo, mas não pode se mover lateralmente. A modelagem cinemática desses robôs é mais complexa, exigindo o cálculo das velocidades das rodas necessárias para alcançar uma determinada trajetória no espaço.

4 Modelo Cinemático de Robôs de Tração Diferencial

O modelo cinemático de um robô de tração diferencial descreve como as velocidades das rodas do robô se relacionam com sua velocidade linear e angular. Este modelo é fundamental para o controle preciso do movimento do robô e é frequentemente usado em algoritmos de navegação e planejamento de trajetória.

5 Implementação das Equações Cinemáticas

Nesta seção, detalharemos como as equações cinemáticas discutidas na seção 2 foram implementadas em código, utilizando a API remota do CoppeliaSim. Esta implementação é essencial para controlar o movimento do robô de tração diferencial na simulação.

5.1 Equações Utilizadas

As equações cinemáticas para um robô de tração diferencial são:

$$v = \frac{V_r + V_l}{2} = r \cdot \frac{\omega_r + \omega_l}{2}$$

$$\omega = \frac{V_r - V_l}{L} = r \cdot \frac{\omega_r - \omega_l}{L}$$

Para a cinemática inversa:

$$\omega_r = \frac{2v + \omega L}{2r}$$

$$\omega_l = \frac{2v - \omega L}{2r}$$

5.2 Implementação em Python

A seguir, detalhamos a implementação das equações em Python, utilizando a API remota do CoppeliaSim para controlar o robô.

5.2.1 Configuração da API Remota

Primeiramente, precisamos configurar a API Remota para estabelecer a comunicação entre o Python e o CoppeliaSim.

Figura 2: Configuração da API remota

```
1  from coppeliasim_zmqremoteapi_client import RemoteAPIClient
2  import numpy as np
3
4  client = RemoteAPIClient()
5  sim = client.getObject("sim")
6
7  # Função de normalização dos ângulos ao intervalo [-pi, pi]
8  def normalizeAngle(angle):
9      |   return np.mod(angle+np.pi, 2*np.pi) - np.pi
10
11 # Handle do P3DX
12 handle_robo = sim.getObject('/Carriim')
```

5.2.2 Controle das Velocidades das Rodas

Utilizando as equações da cinemática inversa, podemos calcular V_r e V_l com base na velocidade linear V e angular ω desejadas.

Figura 3: Código do Controle das Velocidades das Rodas

```

18 # Parametros do robô
19 L = 0.19
20 r = 0.0325
21
22 def calcular_velocidade(v, omega):
23     omega_r = (2*v + omega * L) / (2*r)
24     omega_l = (2*v - omega * L) / (2*r)
25     return omega_r, omega_l
26
27 # Velocidades desejadas
28 v_desejada = 0.5 # Velocidade linear desejada
29 omega_desejada = 0.1 # Velocidade angular desejada
30
31 omega_r, omega_l = calcular_velocidade(v_desejada, omega_desejada)
32
33
34 # definir a velocidade dos motores do Carrim
35 sim.setJointTargetVelocity(roda_direita, omega_r)
36 sim.setJointTargetVelocity(roda_esquerda, omega_l)
37

```

5.2.3 Controle do Robô na Simulação

Após calcular e enviar as velocidades das rodas, o robô se moverá conforme as especificações do modelo cinemático. O controle contínuo pode ser implementado dentro de um loop, ajustando as velocidades conforme necessário.

Figura 4: código do Controle do robô na Simulação

```

31 sim.startSimulation()
32
33 # Loop de controle
34 while sim.getSimulationTime() < 20:
35     # Atualizar velocidades das rodas conforme necessário
36     omega_r, omega_l = calcular_velocidade(v_desejada, omega_desejada)
37
38     # Enviar comandos atualizados para as rodas
39     sim.setJointTargetVelocity(roda_direita, omega_r)
40     sim.setJointTargetVelocity(roda_esquerda, omega_l)
41
42
43
44 # definir a velocidade dos motores do Carrim
45 sim.setJointTargetVelocity(roda_direita, omega_r)
46 sim.setJointTargetVelocity(roda_esquerda, omega_l)
47
48 # Fechar a conexão com o CoppeliaSim
49
50 sim.stopSimulation()
51

```

Descrevemos como as equações cinemáticas são implementadas em Python, utilizando a API Remota do CoppeliaSim para controlar as velocidades das rodas do robô de tração diferencial.

A aplicação desse modelo cinemático permite que o robô execute movimentos precisos e controlados durante a simulação.

CONCLUSÃO

Neste relatório, abordamos a implementação do controle cinemático de um robô de tração diferencial no CoppeliaSim. Exploramos os conceitos teóricos relacionados ao controle cinemático, como a cinemática direta e inversa, e sua aplicação prática na simulação robótica.

- **Principais Pontos Abordados**

- **Controle Cinemático de Robôs Móveis:** Compreendemos a importância do controle cinemático na navegação precisa de robôs móveis.
- **Cinemática Direta e Inversa:** Exploramos as equações que relacionam as velocidades das rodas do robô com sua velocidade linear e angular.
- **Modelagem Cinemática de Robôs de Tração Diferencial:** Implementamos o modelo cinemático para esse tipo de robô, que é não-holonômico.

- **Dificuldades Enfrentadas**

Durante a implementação, algumas dificuldades foram encontradas, tanto no código quanto nas equações cinemáticas. Uma das dificuldades foi a correta configuração dos parâmetros físicos no CoppeliaSim, o que influencia diretamente no comportamento do robô durante a simulação.

- **Aprendizados e Considerações Finais**

Este estudo em robótica móvel e controle cinemático, apesar dos obstáculos, foi enriquecedor. Utilizando o CoppeliaSim, foi possível entender os desafios do controle robótico e aplicar equações cinemáticas para materializar os conceitos teóricos. A experiência contribuiu significativamente para o conhecimento prático na área.