

Parábola (Imersão)

Esta é uma função chamada arcRotation que utiliza vários parâmetros para controlar o movimento de um robô em um caminho curvo.

Aqui está um passo a passo do código:

A função leva vários parâmetros, incluindo:

- Raio
- Ângulo
- Velocidade inicial
- Velocidade máxima
- Velocidade final
- Velocidade adicional
- BrakeStart
- StopMethod
- Gerador
- Stop

1. Booleano

```
if cancel:  
    return  
  
global run_generator, runSmall
```

Ele verifica se uma variável global cancel está definida e, em caso afirmativo, imprime "cancelar" e retorna da função.

Ele define duas variáveis globais run_generator e runSmall como False.

```
if generator == None:  
    run_generator = False
```

Se o parâmetro do gerador for None, ele define run_generator como False.

2. Calibração

```
angle = angle * (2400/2443) #gyro calibration

gyroStartValue = getGyroValue()
finalGyroValue = gyroStartValue + angle
currentAngle = gyroStartValue

accelerateDistance = abs(angle * addspeed)
deccelerateDistance = abs(angle * (1 - brakeStart))
brakeStartValue = abs(angle * brakeStart)

loop = True

#Calculating the speed ratios based on the given radius
if angle * startspeed > 0:
    speed_ratio_left = (radius+14) / (radius+2) #calculate speed
    speed_ratio_right = 1
else:
    speed_ratio_left = 1
    speed_ratio_right = (radius+14) / (radius+2)
```

Ajusta o parâmetro do ângulo com base no valor de calibração do giroscópio. Ele registra o valor inicial do giroscópio e calcula o valor final do giroscópio com base no ângulo ajustado.

Calcula vários valores de distância com base no ângulo e em vários parâmetros de velocidade.

Ele define um loop de flag como True para iniciar um loop infinito.

Calcula as relações de velocidade para os motores esquerdo e direito com base no raio da curva.

Ele calcula a velocidade inicial para os motores esquerdo e direito usando uma função auxiliar speedCalculation.

Ele inicia um loop infinito que continua até que o sinalizador de loop seja definido como False.

3. Stop

```
(angle / abs(angle))
if finalGyroValue * (angle / abs(angle)) < currentAngle * (angle / abs(angle)):
    #print("finalGyroValue: " + str(finalGyroValue) + " rotatedDistance: " + str(currentAngle))
    loop = False
    break
```

Dentro do loop, ele verifica se o botão cancelar está pressionado. Nesse caso, ele sai do loop.