

## Relatório do Laboratório 4 - Otimização com Métodos Baseados em População

### 1 Breve Explicação em Alto Nível da Implementação

O objetivo deste laboratório é aplicar um algoritmo de otimização baseado em população chamado *Particle Swarm Optimization* (PSO), inspirado no comportamento coletivo de bandos de pássaros e cardumes de peixes durante movimentos de busca por alimento ou migração.

#### 1.1 *Particle Swarm Optimization*

O algoritmo PSO trabalha com um conjunto de partículas, cada uma representando uma solução candidata no espaço de busca. A cada iteração, as partículas ajustam suas posições com base em suas próprias experiências (melhor posição individual) e na experiência coletiva do grupo (melhor posição global). A dinâmica do movimento é determinada por uma equação de atualização de velocidade, que incorpora tanto fatores cognitivos quanto sociais.

Inicialmente, as partículas são distribuídas aleatoriamente dentro dos limites do espaço de busca, com posições e velocidades também inicializadas aleatoriamente. A atualização da velocidade de cada partícula segue a seguinte equação:

$$\mathbf{V}_i = \omega \mathbf{V}_i + \varphi_p r_p (\mathbf{b}_i - \mathbf{x}_i) + \varphi_g r_g (\mathbf{b}_g - \mathbf{x}_i) \quad (1)$$

Nessa equação,  $\mathbf{V}_i$  representa a velocidade da partícula  $i$ , enquanto  $\mathbf{x}_i$  corresponde à sua posição atual no espaço de busca. O termo  $\mathbf{b}_i$  refere-se à melhor posição já encontrada pela própria partícula  $i$ . Já  $\mathbf{b}_g$  representa a melhor posição encontrada por todo o enxame até o momento. O parâmetro  $\omega$ , conhecido como *inertia weight*, controla o quanto da velocidade anterior da partícula influencia seu movimento atual, ajudando a balancear entre *exploration* e *exploitation*. Os coeficientes  $\varphi_p$  e  $\varphi_g$  são, respectivamente, os parâmetros cognitivo e social, responsáveis por ponderar a influência das melhores posições individuais e globais. Por fim,  $r_p$  e  $r_g$  são variáveis aleatórias com distribuição uniforme no intervalo  $[0, 1]$ , geradas a cada iteração para introduzir estocasticidade no processo de atualização das velocidades.

A seguir, é apresentado o pseudocódigo do algoritmo PSO utilizado neste experimento:

```
# Assuming minimization
def pso(J, hyperparams):
    particles = initialize_particles(
        hyperparams.num_particles,
        hyperparams.lb,
        hyperparams.ub
    )

    best_global = None # J(None) = inf assumido

    while not check_stopping_condition():
        particles, best_iteration = update_particles(
            particles, best_global, hyperparams
        )

        if best_global is None or J(best_iteration) < J(best_global):
            best_global = best_iteration

    return best_global
```

## 2 Figuras Comprovando Funcionamento do Código

### 2.1 Teste do *Particle Swarm Optimization*

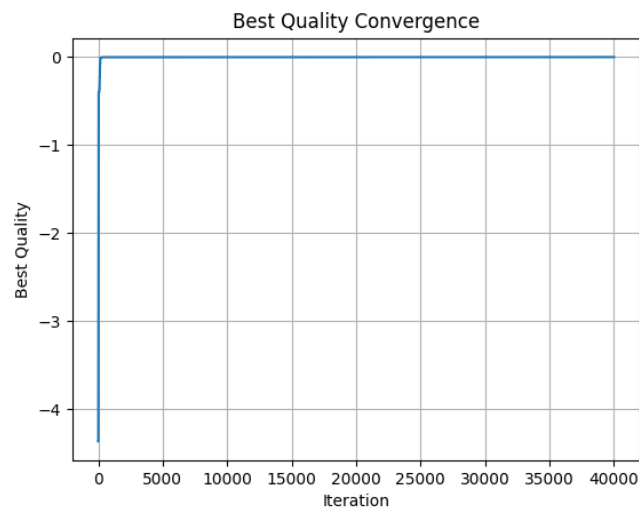


Figura 1: gráfico da convergência da melhor qualidade no caso teste.

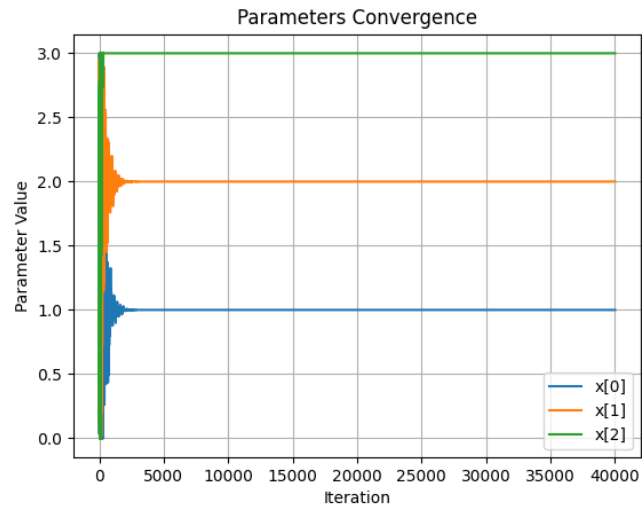


Figura 2: gráfico da convergência dos parâmetros no caso teste.

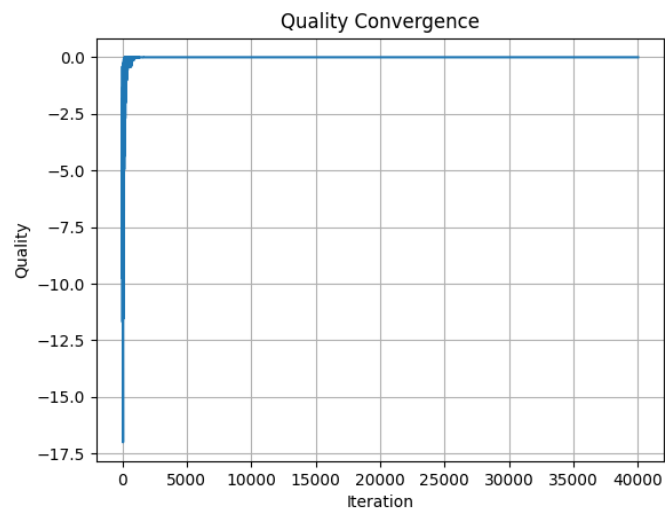


Figura 3: gráfico da convergência da qualidade no caso teste.

## 2.2 Otimização do controlador do robô seguidor de linha

### 2.2.1 Histórico de Otimização

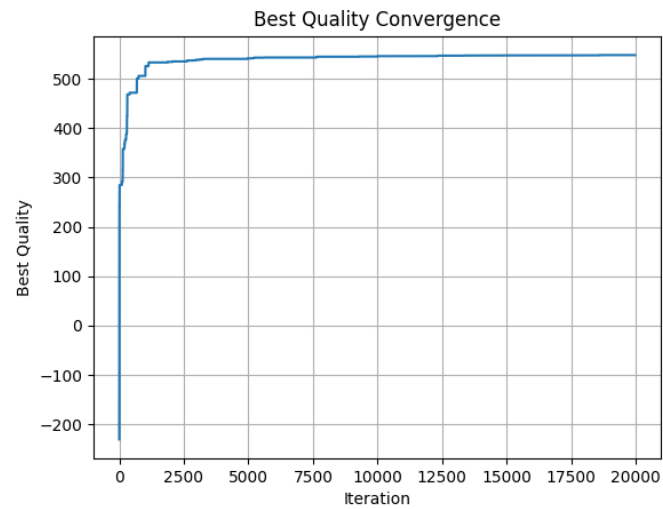


Figura 4: gráfico da convergência da melhor qualidade para o seguidor de linha.

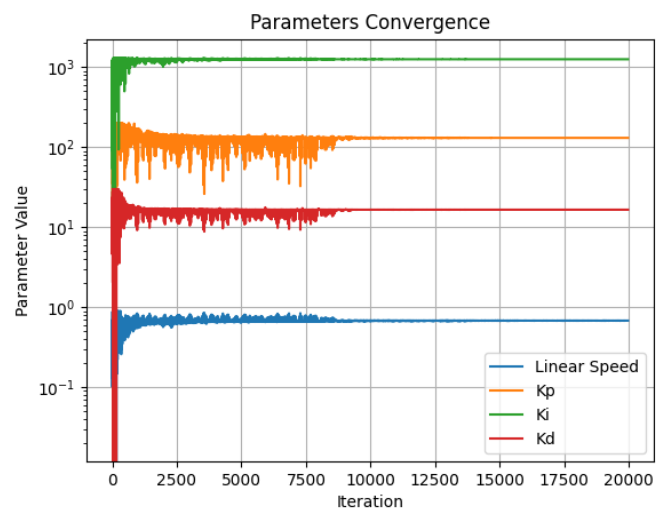


Figura 5: gráfico da convergência dos parâmetros para o seguidor de linha.

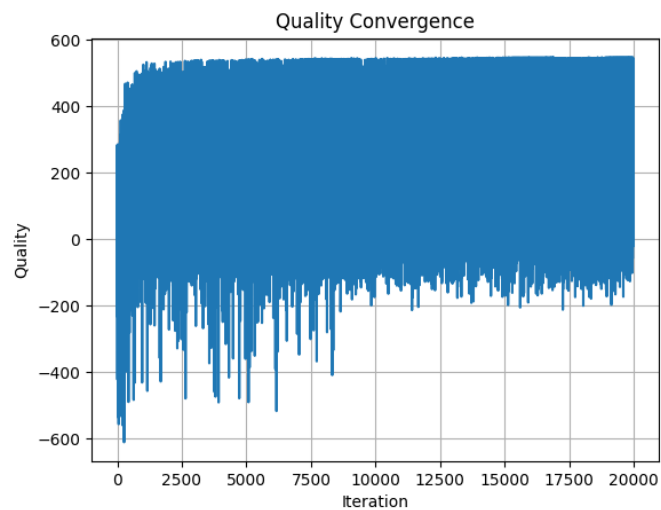


Figura 6: gráfico da convergência da qualidade para o seguidor de linha.

### 2.2.2 Melhor Trajetória Obtida Durante a Otimização

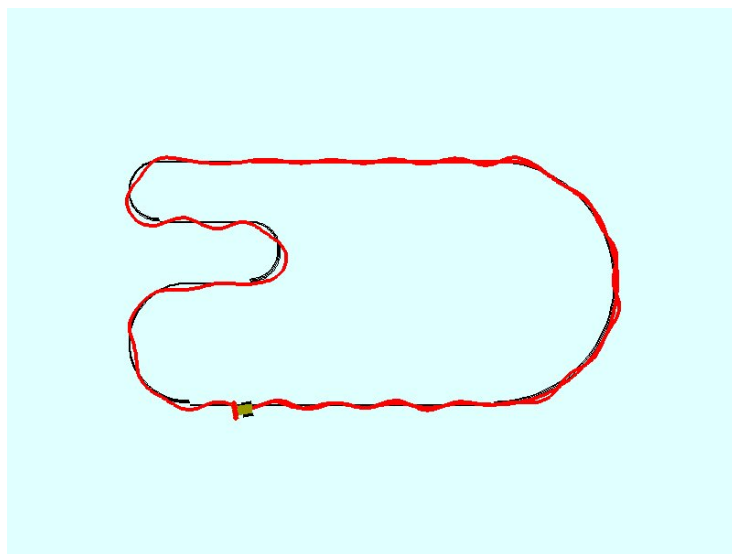


Figura 7: Melhor caminho encontrado para o robô seguidor de linha.

### 3 Discussão sobre o observado durante o processo de otimização

O processo de otimização com aprendizado acelerado levou aproximadamente 30 minutos para atingir resultados satisfatórios. Realizei testes com diferentes durações de otimização, incluindo períodos de até 2 horas, que mesmo assim os resultados não haviam convergido por completo. Os resultados encontrados com maior tempo de otimização eram bem próximos dos melhores valores encontrados com 30 minutos de otimização. Os valores encontrados para os parâmetros e para a função de qualidade para os gráficos do relatório foram os seguintes:

$$\mathbf{params:} [0.681842, 130.960183, 1257.179981, 16.581492], \mathbf{quality:} 548.0256527863156 \quad (2)$$