

Inteligência Artificial para Robótica Móvel

CT-213

Instituto Tecnológico de Aeronáutica

Relatório do Laboratório 4 - Otimização com Métodos Baseados em População

Leonardo Peres Dias

1 de maio de 2025



Sumário

1 Breve explicação em alto nível da implementação	3
1.1 Particle Swarm Optimization	3
2 Figuras comprovando funcionamento do código	4
2.1 Teste do Particle Swarm Optimization	4
2.2 Otimização do controlador do robô seguidor de linha	5
2.2.1 Histórico de otimização	5
2.2.2 Melhor trajetória obtida durante a otimização	6
3 Discussão sobre o observado durante o processo de otimização	6

1 Breve explicação em alto nível da implementação

1.1 Particle Swarm Optimization

A implementação do algoritmo de PSO foi dividida em duas principais classes: `Particle`, que representa uma partícula individual do enxame, e `ParticleSwarmOptimization`, que gerencia o comportamento coletivo do enxame e executa o algoritmo.

Classe `Particle`

Cada instância da classe `Particle` corresponde a uma partícula que se movimenta dentro do espaço de busca definido pelos limites `lower_bound` e `upper_bound`. No momento da criação, a posição x é inicializada com valores aleatórios dentro desses limites. A velocidade v também é inicializada com valores aleatórios, porém dentro do intervalo simétrico definido pelo vetor $\Delta = u_b - l_b$.

A partícula mantém ainda duas variáveis adicionais: a melhor posição já visitada pela partícula (`bi`) e o valor da função objetivo nessa posição (`bi_fitness`). Inicialmente, `bi` é a posição de origem da partícula e `bi_fitness` é definido como $-\infty$ para garantir que qualquer valor avaliado inicialmente o substitua.

Classe `ParticleSwarmOptimization`

A classe `ParticleSwarmOptimization` gerencia a população de partículas e implementa as etapas do algoritmo PSO. Recebe como entrada os hiperparâmetros do algoritmo (`inertia_weight`, `cognitive_parameter`, `social_parameter`) e os limites do espaço de busca. No construtor, é feito o instanciamento de todas as partículas, e os atributos `bg` e `bg_fitness` são inicializados para armazenar, respectivamente, a melhor posição global e seu valor associado.

O método `get_position_to_evaluate` retorna sequencialmente a posição de cada partícula para ser avaliada externamente. Após a avaliação, o método `notify_evaluation` é chamado, atualizando a melhor posição pessoal da partícula e, se for o caso, também a melhor posição global do enxame.

Quando todas as partículas da geração atual tiverem sido avaliadas, o método `advance_generation` é executado. Ele realiza a atualização das velocidades de todas

as partículas utilizando a equação padrão do PSO:

$$v \leftarrow wv + \phi_p r_p (b_i - x) + \phi_g r_g (b_g - x)$$

onde w é o fator de inércia, ϕ_p e ϕ_g são os parâmetros cognitivo e social, e r_p e r_g são variáveis aleatórias uniformemente distribuídas em $[0, 1]$. Após o cálculo da nova velocidade, ela é limitada ao intervalo máximo admissível (v_{max}), a nova posição da partícula é calculada $x_i = x_i + v_i$ e restringida aos limites do espaço de busca.

2 Figuras comprovando funcionamento do código

2.1 Teste do Particle Swarm Optimization

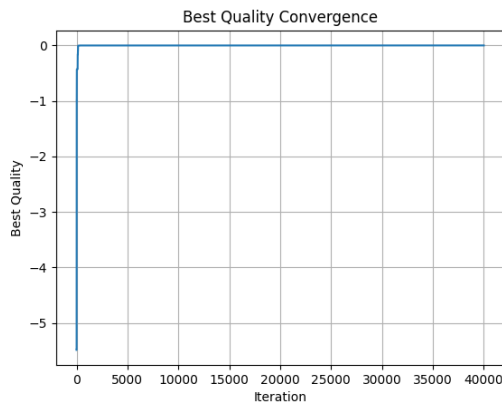


Figura 1: Evolução do melhor fitness global ao longo das gerações.

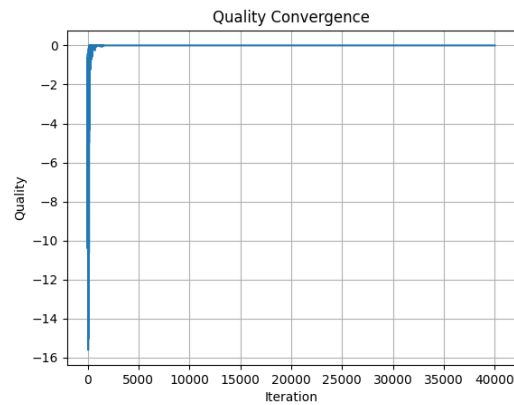


Figura 2: Evolução do fitness das partículas ao longo das gerações.

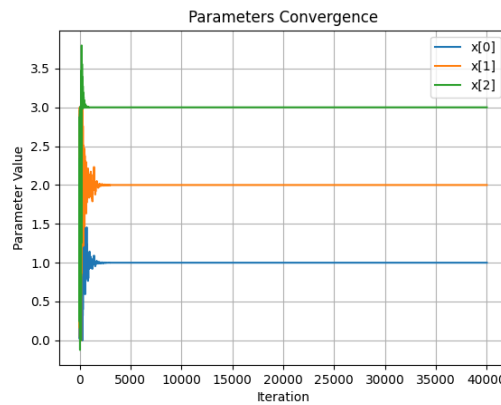


Figura 3: Evolução dos parâmetros.

2.2 Otimização do controlador do robô seguidor de linha

2.2.1 Histórico de otimização

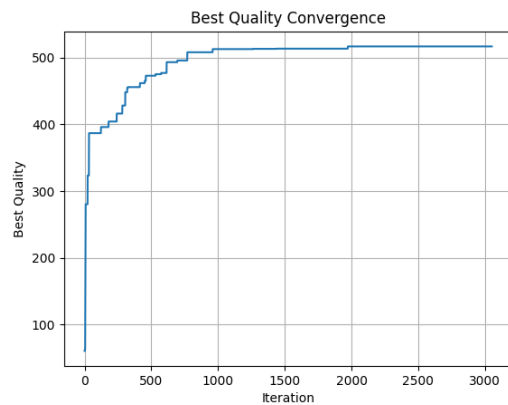


Figura 4: Evolução do melhor fitness global ao longo das gerações.

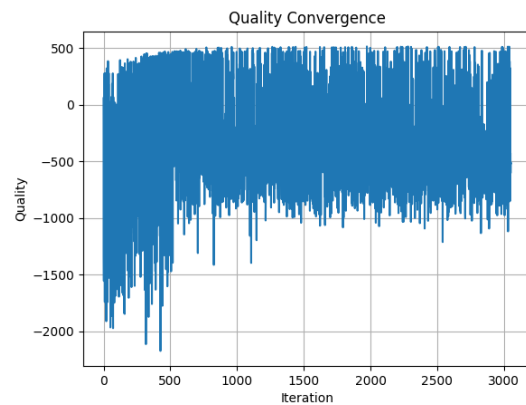


Figura 5: Evolução do fitness das partículas ao longo das gerações.

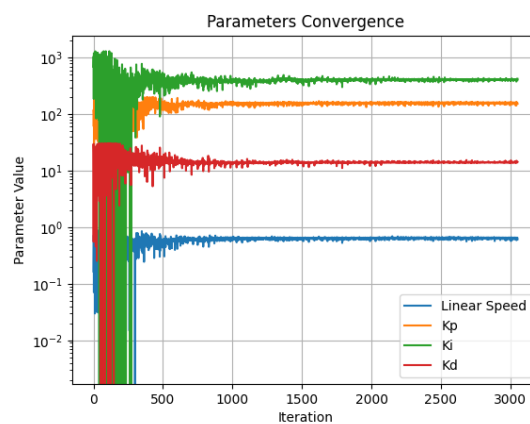


Figura 6: Evolução dos parâmetros.

2.2.2 Melhor trajetória obtida durante a otimização

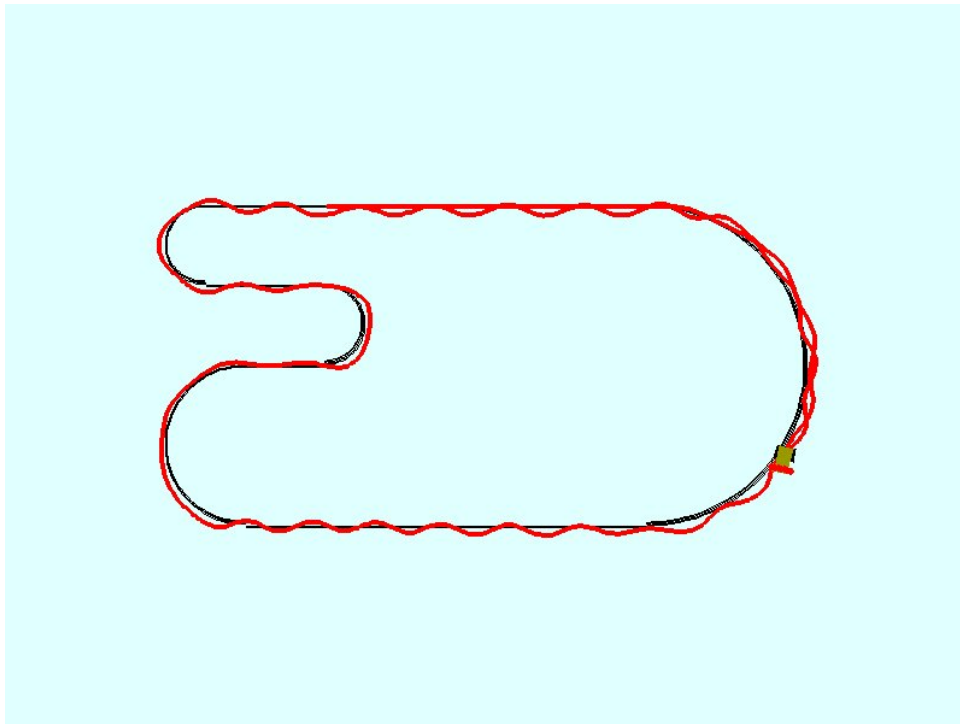


Figura 7: Melhor trajetória obtida durante a otimização.

3 Discussão sobre o observado durante o processo de otimização

Durante o processo de otimização, observou-se que o algoritmo PSO não convergiu totalmente para o máximo global, porém conseguiu encontrar um conjunto de parâmetros que nos fornece um bom desempenho do seguidor de linha. A partir da análise das figuras apresentadas, é possível notar que a qualidade de cada uma das partículas não se estabilizou como se observa no `test_pso`, mas a melhor qualidade global sim. Nesse sentido, nota-se que para o problema do seguidor de linha, quando se para o treinamento, não temos ainda todas as partículas posicionadas em um máximo global.