Instituto Tecnológico de Aeronáutica - ITA
Inteligência Artificial para Robótica Móvel - CT213
Aluno: Reynaldo Santos de Lima

Relatório do Laboratório 4 - Otimização com Métodos Baseados em População

1 Breve Explicação em Alto Nível da Implementação do *PSO*

1.1 Implementação do *Particle Swarm Optimization*

A implementação do PSO seguiu o apresentado em sala. Como no caso do laboratório se tratava de um problema de maximização, alterou-se as comparações. Além disso, este trecho do código é implementado no método *notify_ evaluation*, que recebe o valor da função qualidade, visto que esta é externa ao PSO. Este trecho do código foi implementado como a seguir:

```
def notify_evaluation(self, value):
        if value > self.particles[self.current_particle].best_value:
                self.particles[self.current_particle].best =
                                self.particles[self.current_particle].x
                self.particles[self.current_particle].best_value = value
        if value > self.best_iteration_value:
                self.best_iteration = self.particles[self.current_particle].x
                self.best_iteration_value = value
        self.current_particle += 1
        if self.current_particle == self.num_particles:
                self.advance_generation()
```

Observa-se que as comparações seguem a lógica do PSO proposto, mudando apenas a desigualdade. O método *advance_ generation*, por sua vez, segue o final do método proposto em aula, passo que seria após o método genérico update:

```
def advance_generation(self):
        self.current_particle = 0
        if self.best_iteration_value > self.best_global_value:
                self.best_global = self.best_iteration
                self.best_global_value = self.best_iteration_value
```

A atualização das posições em si são feitas no método *notify_ evaluation*. Como particularidade para este, tem-se apenas a previsão do caso da primeira iteração:

```
def get_position_to_evaluate(self):
        if self.best_global is None:
                return self.particles[self.current_particle].x
        else:
                # segue implementação idêntica ao proposto em aula
```

1.2 Função de qualidade

Por fim, como a função qualidade é feita somando-se a recompensa em cada tempo de iteração, fez-se o método *evaluate*. Deste, como particularidade, destaca-se apenas a escolha de uma punição alta para o caso em que não se detecta a linha no sensor. Fez-se erro como $e = 2$ neste caso:

```
def evaluate(self):
        # Escolha dos parâmetros como proposto em aula
        w = 0.5
        if not detection:
               reward = linear * dot_product - w * 2
                return reward
        else:
                reward = linear * dot_product - w * abs(error)
        return reward
```

2 Figuras Comprovando Funcionamento do Código

2.1 Teste do *Particle Swarm Optimization*

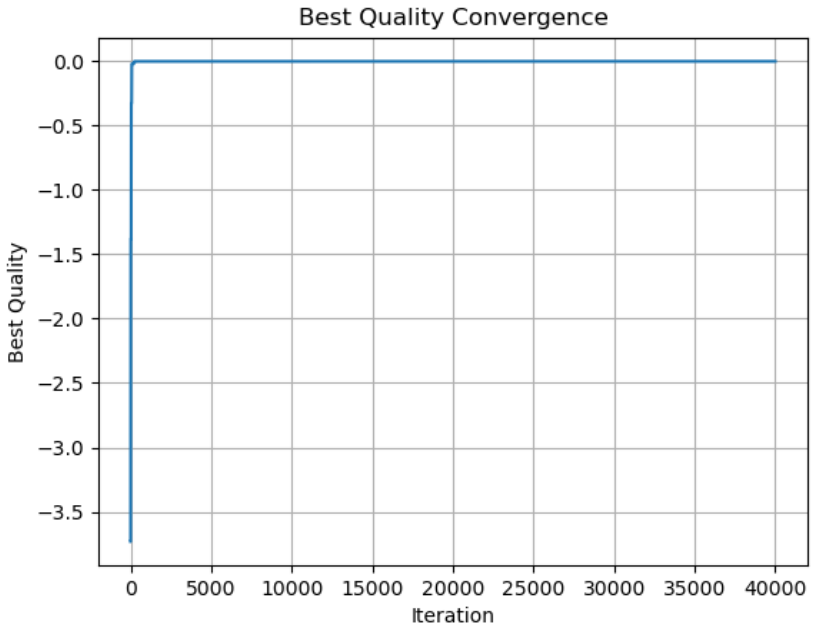

Figura 1: Histórico da variação da melhor solução.

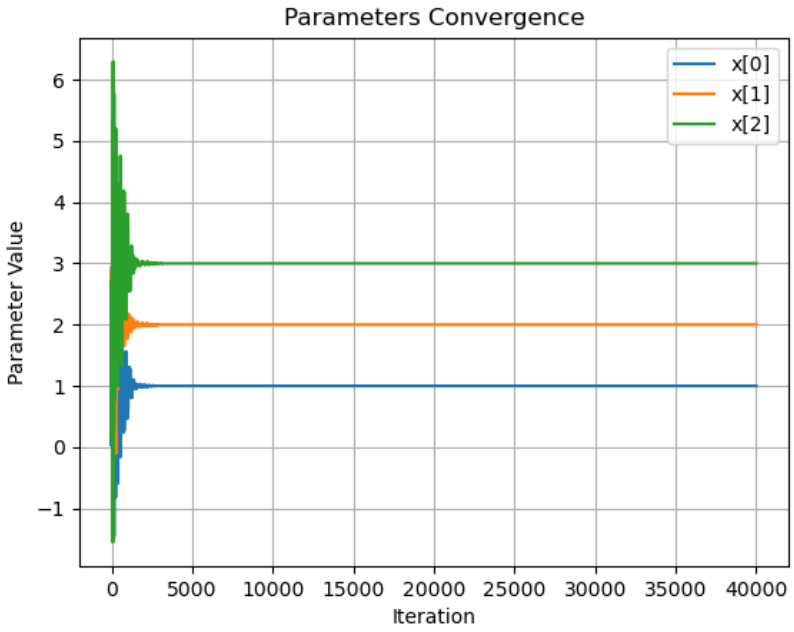

Figura 2: Histórico da escolha dos parâmetros.

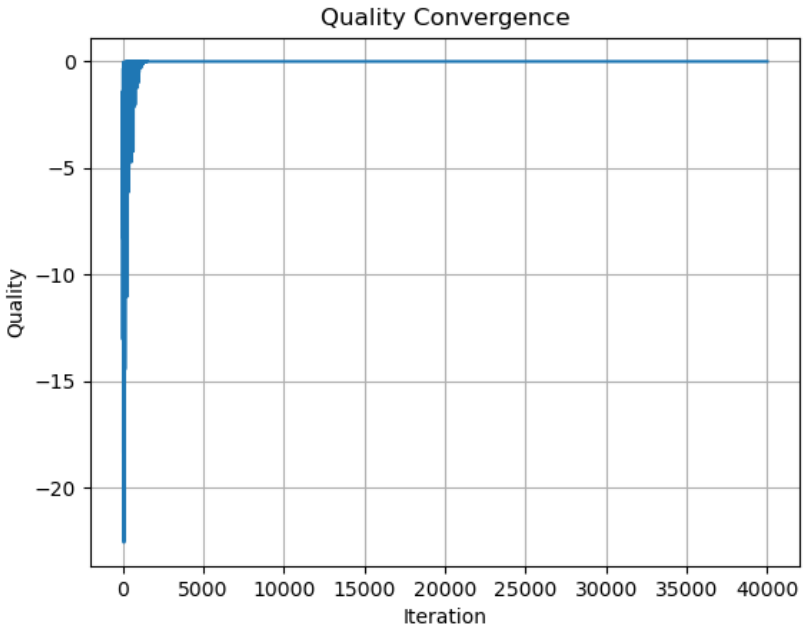

Figura 3: Histórico da função qualidade por iteração.

2.1.1 Histórico de Otimização

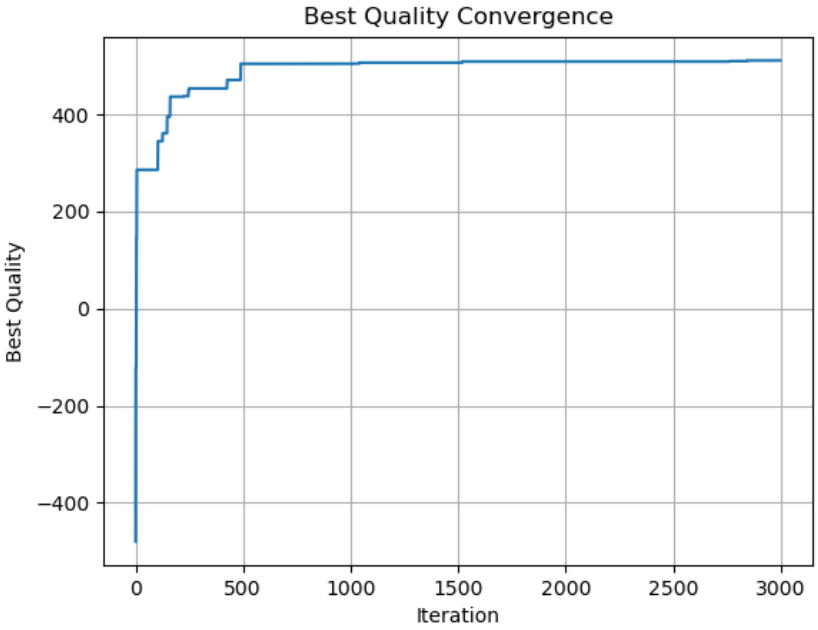

Figura 4: Histórico da variação da melhor solução.

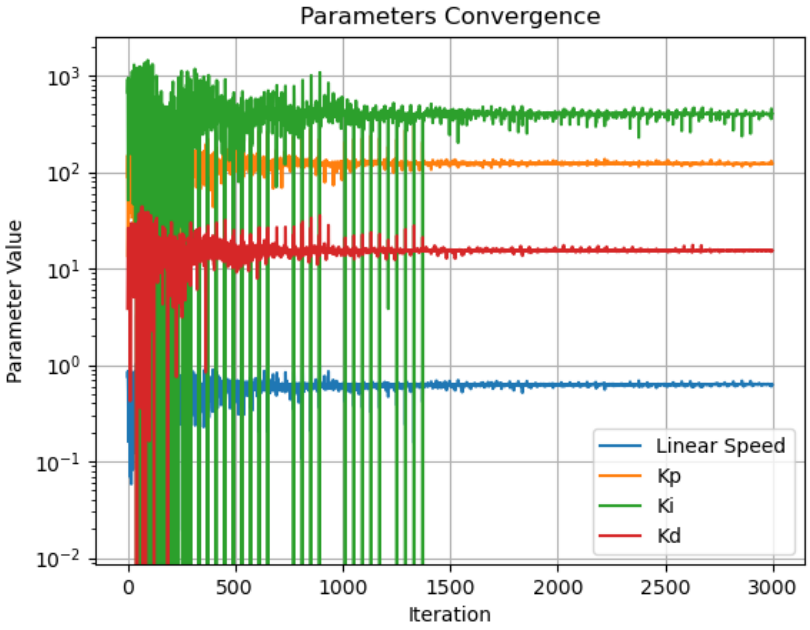

Figura 5: Histórico da escolha dos parâmetros do controlador.

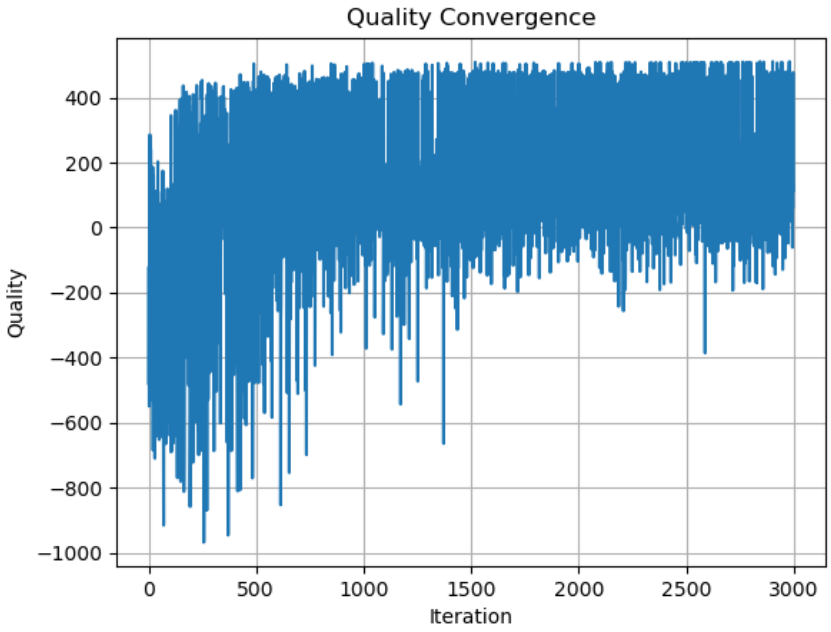

Figura 6: Histórico da função qualidade por iteração.

2.1.2 Melhor Trajetória Obtida Durante a Otimização

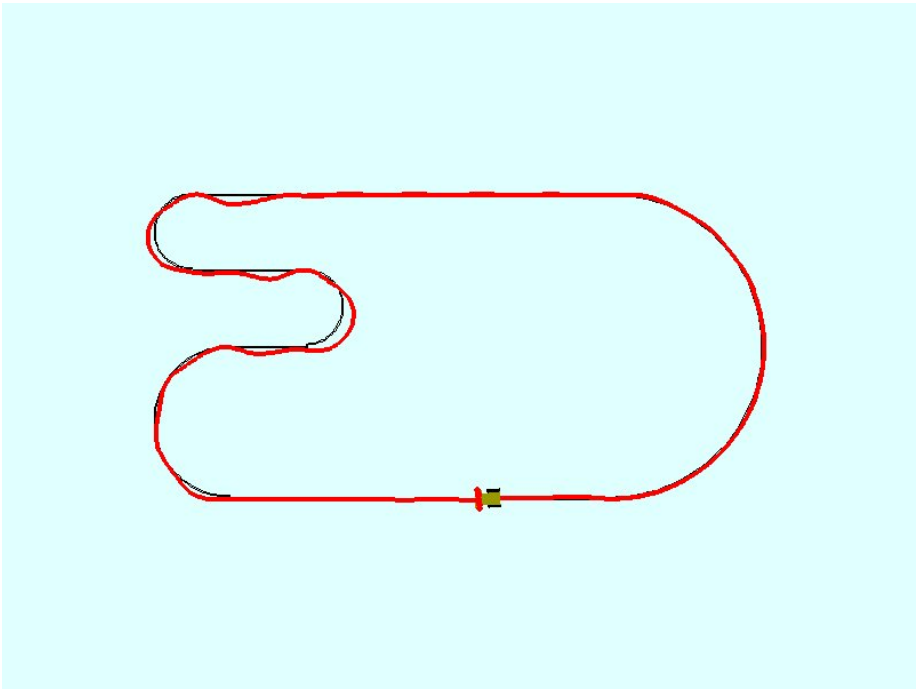

Figura 7: Melhor trajetórica obtida.

3 Discussão sobre o observado durante o processo de otimização

Durante o processo de otimização foi observado como o otimizador faz variações em torno de pontos de solução local, com algum nível de exploração. Ao testar o uso do hiperparâmetro de punição para o caso de não detectar a bola, foi claro como aumentar esta punição tendia para uma solução mais rápido.

Além disso, vale destacar o nível de exploração mencionado; no começo fica mais claro como cada partícula está explorando e não tem uma noção do melhor caso. Com o passar do tempo, ao observar as iterações, tem-se a tendência de algumas partículas (já mais próximas de um máximo local) de ter baixa alteração da sua posição, o que fica em contraste com as demais partículas que exploram mais. Esta tendência segue o esperado pelo proposto para o método.