Instituto Tecnológico de Aeronáutica

CT-213

Laboratório 4 – Otimização com Métodos Baseados em População

Aluno: Pedro Elardenberg Sousa e Souza
Professor: Marcos Ricardo Omena de Albuquerque Maximo

Conteúdo

1 Resumo

Neste Laboratório, foi implementada uma otimização dos parâmetros do controlador de um robô seguidor de linha usando *Particle Swarm Optimization* (PSO).

Os resultados foram mostrados e discutidos neste relatório.

2 Introdução

O problema que se pretende resolver é a otimização do controlador de um robô seguidor de linha. O controlador mantém velocidade linear constante, enquanto utiliza um controlador PID para escolher a velocidade angular:

$$\omega = K_p e + K_i \int ede + K_d e' \tag{1}$$

Em que ω é a velocidade angular comandada e e é o erro de seguimento da linha. Além disso, K_p, K_i e K_d são os ganhos proporcional, integrativo e derivativo, respectivamente. Os parâmetros sujeitos a otimização são a velocidade linear e os 3 ganhos do controlador, logo são 4 parâmetros.

Como medida de qualidade, utilizou-se a seguinte equação:

$$f(x) = \sum_{k=1}^{N} reward_k \tag{2}$$

Em que $reward_k$ é a "recompensa" obtida pelo robô no instante k e N é a duração do episódio de treinamento em passos de tempo *(time steps)*. Para o cálculo da recompensa, utilizou-se:

$$reward_k = v_k \times (r_k, t_k) - w \times e_k V \tag{3}$$

Em que v_k é a velocidade linear (projetada no eixo local do robô) executada pelo robô no instante k, r_k é um vetor (bidimensional) unitário que aponta na direção do robô no instante k, t_k é o vetor tangente à atual posição no caminho no instante k, $e_k V$ é o módulo do erro em relação à linha e w é um peso para fazer um compromisso entre se manter no centro da linha e seguir o caminho rapidamente.

Essa medida de qualidade recompensa o robô por cumprir o caminho mais rapidamente, enquanto o penaliza por desviar da linha. O termo (r_k, t_k) significa o produto interno entre r_k e t_k, e foi adicionado para penalizar o robô caso ele se mova em reverso, pois sem esse termo o PSO às vezes converge para uma solução que o robô faz o circuito (ou uma parte dele) em reverso (ao contrário). Quando o robô está perfeitamente alinhado com a linha, tem-se $(r_k, t_k) = 1$, enquanto quando ele se move em reverso, tem-se $(r_k, t_k) = -1$.

2.1 *Particle Swarm Optimization*

O modelo baseado em população *Particle Swarm Optimization* (PSO) [1] guarda a melhor posição já encontrada para cada partícula b_i, bem como

a melhor posição geral já encontrada b_g e a atualização da velocidade da partícula, ou seja, o quanto ela deve se mover e para que direção, é dada por:

$$v_i = \omega v_i + \phi_p r_p(b_i - x_i) + +\phi_g r_g(b_g - x_i) \tag{4}$$

Em que:

- ω é o "peso"de inércia. O quanto a partícula vai caminhar naquela direção;
- ϕ_p é o parâmetro cognitivo. O quanto a partícula vai caminhar na direção da melhor posição que a partícula b_i já encontrou;
- ϕ_g é o parâmetro social. O quanto a partícula vai caminhar na direção da melhor posição que o algoritmo já encontrou;
- $r_p, r_g \sim U([0, 1])$.

A ideia do algoritmo é orientar cada partícula na direção dos melhores resultados já obtidos.

3 Análise dos Resultados

Os métodos implementados foram os seguintes:
Das classes de *particle_swarm_optimization.py:*

- Particle *__init__()*: inicializa posição e velocidade inicial da partícula, cria os valores para o melhor custo encontrado;
- PSO *__init__()*: inicializa o enxame com o número de partículas definido;
- PSO *notify_evaluation()*: notifica ao algoritmo que a posição da partícula foi avaliada.

Da classe *simulation.py:*

- *evaluate()*: avalia o valor de $reward_k$ dado pela equação 3

No caso de teste, o problema específico a ser resolvido é a determinação do coeficiente de desaceleração de uma bola em movimento num campo de futebol de robôs. A bola em movimento perde energia devido a um fenômeno conhecido como *rolling friction*. Pode-se determinar o coeficiente de desaceleração seguindo os seguintes passos:

O programa convergiu em cerca de 1500 iterações, como mostra a figura 1 e chegou-se ao resultado da trajetória do robô indicada pela figura 2.

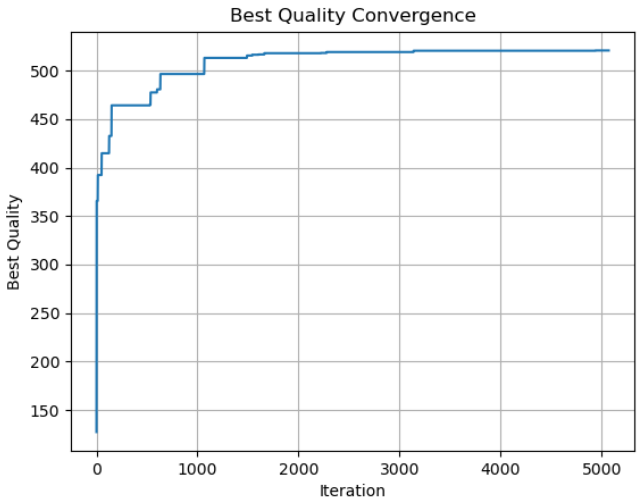

Figura 1: Convergência do melhor valor global

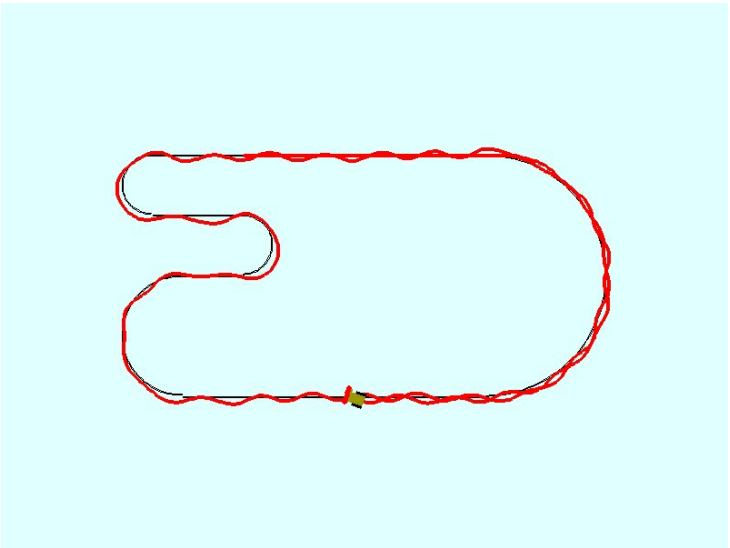

Figura 2: Trajetória feita pelo robô

Os parâmetros otimizados por esse algoritmos são mostrados na figura 3.

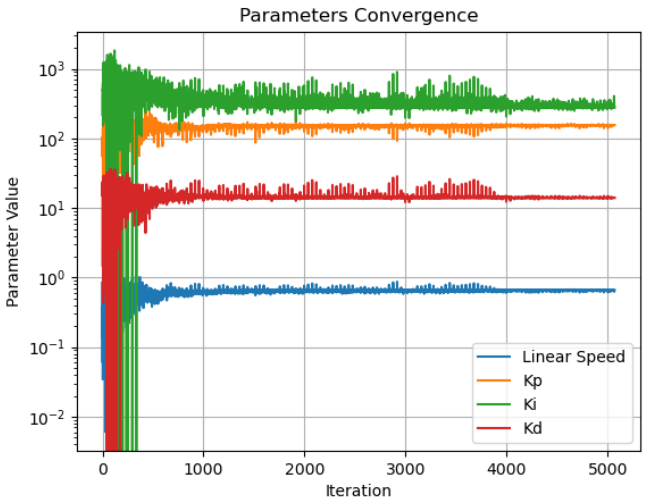

Figura 3: Convergência dos parâmetros otimizados pelo algoritmo

4 Conclusão

O presente laboratório mostrou que o algoritmo de otimização com métodos baseado em população *Particle Swarm Optimization* é eficaz para otimização de parâmetros de aprendizado de máquina. Como se trata de um problema mais genéricos. Com 4 parâmetros a serem otimizados, o método aproximou-se bastante da trajetória ótima. Contudo, foi necessário rodar mais iterações no código e aguardar por mais tempo até a sua convergência.

Referências

[1] Marcos Ricardo Omena de Albuquerque Maximo. Ct-213 - aula 4 - métodos de otimização baseados em população. Apostila, 2020. Curso CT-213 - Inteligência Artificial para Robótica Móvel, Instituto Tecnológico de Aeronáutica.