

Computação Evolutiva aplicada ao Controle de um Manipulador Robótico

Lázaro Pereira Vinaud Neto, Nicolas Carreiro Rodrigues

Eduardo do Valle Simões (Orientador)

ICMC/USP

vinaud@usp.br, nicolascarreiror@usp.br, simoes@icmc.usp.br

Objetivos

O objetivo deste projeto é conduzir um manipulador robótico de até 30 juntas até seu alvo em um ambiente contendo obstáculos complexos como corredores estreitos. Para viabilizar a realização de experimentos, será desenvolvido um software que simula o comportamento de um braço mecânico em duas dimensões utilizando algoritmos evolutivos, tal como proposto por Holland, J. (Holland, 1975), para encontrar configurações ótimas dos ângulos das juntas. A inovação do projeto está em combinar o algoritmo A* com algoritmos evolutivos para ajustar os ângulos das juntas incrementalmente. Este software buscará reduzir a complexidade computacional associada aos algoritmos evolutivos tradicionais, permitindo a obtenção de soluções mais rápidas e eficientes. Juntamente com o desenvolvimento de uma interface gráfica simples e eficaz, para auxiliar o uso do projeto para ensino de computação evolutiva na disciplina SSC 0713 - Sistemas Evolutivos e Aplicados à Robótica, sendo assim essencial uma interface simples. “É importante que os estudantes tenham um modelo bom o suficiente de interface para o usuário desde o começo do aprendizado” (Thimbleby, 1990). Este software pode ser utilizado por alunos que estudam controle de braço robótico, bem como algoritmos evolutivos, pois será disponibilizado sob a licença GPL e como código aberto por meio dos repositórios GitHub

<https://github.com/LVinaud/bracoEvolutivo> e <https://github.com/nicocarreiro/Multi-use-Interface>.

Métodos e Procedimentos

Para enfrentar os desafios de controlar o braço robótico em ambientes com obstáculos complexos, o projeto propõe uma abordagem que combina o algoritmo A* com algoritmos evolutivos. O A* é utilizado para determinar a rota mais otimizada até o objetivo final, dividindo essa rota em waypoints. Esses waypoints servirão como metas intermediárias guiando a evolução incremental do braço robótico. O software foi desenvolvido utilizando C e a biblioteca gráfica SDL, criando um ambiente para a simulação, com a interface utilizando Python, com a biblioteca Tkinter(Tkinter). Para o auxílio do uso do software no ensino, a interface desenvolvida altera as configurações de cada experimento, em parâmetros como o número de juntas, o tamanho da população, o valor da mutação, o tamanho das juntas, etc., através de janelas interativas, além de permitir que esses valores sejam gravados como um cenário para que possam ser carregados novamente depois.

Resultados

Como apresentado na Figura 1, o algoritmo é capaz de encontrar caminhos otimizados, bem como desviar de obstáculos maximizando a

distância da junta mais próxima a qualquer obstáculo em tempo relativamente curto. Uma população é capaz de reduzir a distância ao objetivo em um tempo relativo curto, como apontado pelo gráfico da Figura 2 em uma simulação de 40 gerações com tempo de processamento inferior a um segundo. Como característica de um sistema evolutivo, o algoritmo pode continuar indefinidamente até uma solução desejada, o que permite que o simulador encontre otimizações ainda mais refinadas com um tempo maior de processamento.

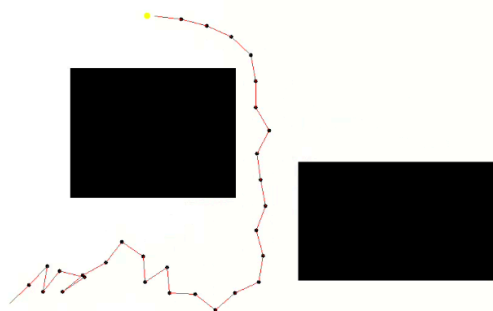


Figura 1: Configuração de um braço robótico a partir da base até o alvo, gerado pelo algoritmo genético.

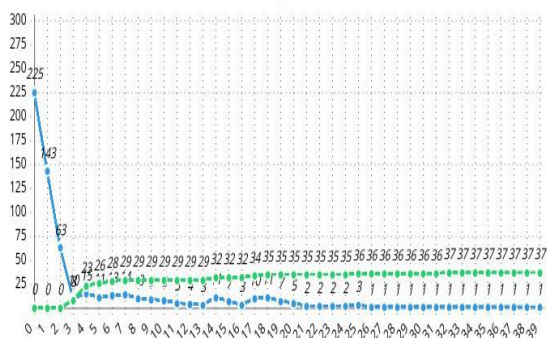


Figura 2: No eixo x, tem-se o número de Gerações. No eixo y, tem-se a distância mínima a um obstáculo em verde e a distância ao objetivo em azul.

Conclusões

O projeto integra técnicas de otimização de trajetórias e evolução genética para

desenvolver um sistema robusto e eficiente de controle de braços robóticos com aplicações na robótica e automação, além de estimular o uso de tecnologias de informação como ferramentas de aprendizagem. A ferramenta auxilia também no ensino de computação bioinspirada a partir de um simulador de fácil uso.

Agradecimentos

Os autores agradecem ao seu orientador Eduardo Simões, e ao Instituto de Ciências Matemáticas e de Computação pela infraestrutura, disponibilidade de laboratórios e apoio teórico para o desenvolvimento do projeto.

Referências

Holland, J. H. (1975). Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. U Michigan Press.

Thimbleby, H. (1990). User Interface Design. ACM Press.

Tkinter - Python interface to Tcl/Tk, disponível em:
<https://docs.python.org/3/library/tkinter.html>,
acessado em 5/6/24.