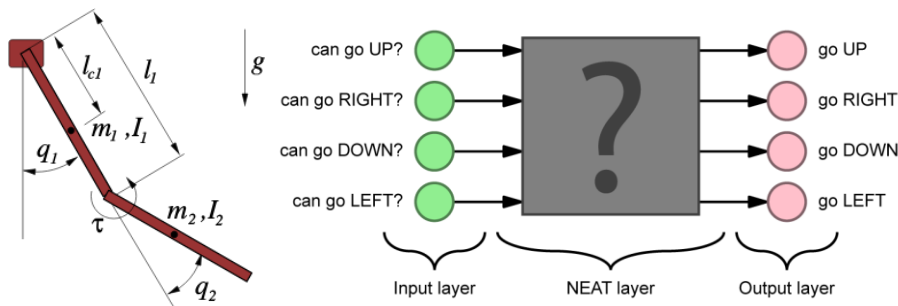


Construcción de un Acrobot con Neurocontrolador NEAT

Jorge Kessler Martín

Objetivo

En este proyecto, se propone desarrollar un sistema de control para un acrobot utilizando un neurocontrolador basado en el algoritmo NEAT (NeuroEvolution of Augmenting Topologies). El acrobot es un sistema dinámico compuesto por dos brazos rígidos conectados por una articulación, donde el objetivo es controlar el movimiento del extremo del segundo brazo. El uso de un neurocontrolador permite lograr un control adaptativo y robusto del sistema, donde se busca que sea capaz de aplicar el torque necesario para que el acrobot de manera autónoma se levante y mantenga el equilibrio.



Herramientas

Para poder realizar este proyecto se van a utilizar las siguientes herramientas:

- CoppeliaSim (simulador de robots): donde se va a realizar la simulación de comportamiento y el entrenamiento necesario
- Python: lenguaje interpretado con multitud de librerías, entre ellas NEAT-Python, con las que se implementará la programación de comportamiento del robot

Metodología

Para realizar el proyecto se realizarán los siguientes pasos:

- Diseñar e implementar un acrobot sencillo en el simulador CopelliaSim
- Realizar un modelo de redes neuronales con NEAT ,desde Python, para aplicar sobre el acrobot creado
- Conectar el código realizado en Python con CopelliaSim mediante las APIs que proporciona el simulador
- Realizar el entrenamiento NEAT, donde se tienen que realizar varias simulaciones donde se prueben diversas redes neuronales hasta que el algoritmo encuentra la mejor topología de red

Estado del arte

En este apartado podemos observar el estado actual del acrobot y de NEAT.

Los enfoques convencionales para el control del acrobot se basaban en métodos de control clásicos como el control proporcional integral derivativo (PID) y el control adaptativo con aprendizaje por refuerzo [1].

El algoritmo de neuroevolución NEAT ha ganado popularidad como un enfoque prometedor para la optimización y el diseño de redes neuronales complejas. NEAT aborda el desafío de la topología de red al permitir la evolución de la estructura de la red, lo que facilita la adaptación y la mejora continua de la arquitectura de la red en función de las demandas específicas del problema [2].

El algoritmo NEAT tiene también una implementación en Python creada sobre 2015, que incluye toda la documentación para poder aplicar esta técnica a cualquier proyecto realizado en este lenguaje [3].

Referencias

- [1] [Evolving Neural Networks through Augmenting Topologies](#), by Kenneth O. Stanley, Risto Miikkulainen, 2002
- [2] [Reinforcement Learning: An Introduction](#), by Richard S. Sutton and Andrew G. Barto, 2018
- [3] [NEAT-Python's Documentation](#), 2015