**Guideline for PPO-Based Antenna Placement Optimization Code**

This document provides a comprehensive guideline for understanding and using the MATLAB code designed to optimize the 3D placement of antennas using Proximal Policy Optimization (PPO). The code leverages MATLAB’s Reinforcement Learning Toolbox to train an agent to maximize Signal-to-Noise Ratio (SNR) and minimize Bit Error Rate (BER) in a simulated indoor environment.

**1. Purpose of the Code**

The code aims to:

* Optimize the placement of an Access Point (AP) and two receivers (Rx) in a 3D room.
* Use PPO to dynamically adjust antenna positions to achieve optimal communication quality.
* Simulate realistic indoor environments with obstacles and material reflection properties.

**2. Code Overview**

**2.1. Key Components**

* **Room and Environment Setup:**
  + Simulates a 4m x 4m x 3m indoor environment.
  + Models objects as obstacles with material properties affecting signal propagation.
* **Reinforcement Learning Environment:**
  + Observations: Antenna positions and environmental features.
  + Actions: Adjustments to antenna positions in 3D space.
  + Rewards: Combines SNR maximization and BER minimization.
* **PPO Agent:**
  + Policy network to determine actions.
  + Value network to estimate state values.
  + Optimized using clipping and advantage functions.
* **Training Process:**
  + Uses the Reinforcement Learning Toolbox to train the PPO agent over multiple episodes.
  + Stops training when the average reward meets the defined threshold.
* **Evaluation:**
  + Predicts optimal antenna placements.
  + Visualizes the environment, obstacles, and antenna positions.

**2.2. Dependencies**

* MATLAB R2021a or later.
* Reinforcement Learning Toolbox.
* Communications Toolbox (optional, for Rician channel simulation).

**3. How to Use the Code**

**3.1. Installation**

1. Ensure MATLAB is installed on your system.
2. Verify that the Reinforcement Learning Toolbox is available.
3. Install any missing toolboxes via MATLAB Add-On Manager.

**3.2. Running the Code**

1. Clone the repository:
2. git clone https://github.com/saeidim/Wifi-Tx-Rx-3D-placement
3. Open MATLAB and navigate to the directory containing the code.
4. Run the script: ppo.m

**3.3. Customizing Parameters**

* Adjust the room dimensions, number of objects, or material reflection coefficients in the initialization section.
* Modify the PPO agent options to experiment with different training configurations.

**3.4. Visualizing Results**

* After training, the script displays a 3D visualization of the room, objects, and antenna placements.
* Evaluate the performance metrics (e.g., SNR and BER) for different configurations.

**4. Code Structure**

**4.1. Main Script**

* Initializes the environment and PPO agent.
* Defines the training loop and visualizes results.

**4.2. Helper Functions**

* resetEnvironment: Resets the environment with randomized object positions.
* stepEnvironment: Simulates actions and computes rewards.
* computeSignalQuality: Calculates SNR and BER for given antenna positions.
* randomizeObjects: Generates random obstacles in the environment.
* extractFeatures: Extracts features for PPO training.

**5. Key Features**

* **Simulated Environment:** Models realistic indoor spaces with configurable obstacles.
* **Reinforcement Learning Integration:** Fully utilizes MATLAB’s RL Toolbox.
* **Visualization:** Provides intuitive 3D plots of the room, obstacles, and optimized antenna placements.
* **Modular Design:** Easy to extend or customize for other reinforcement learning algorithms.

**6. Suggested Improvements**

* Incorporate a more sophisticated reward function considering additional metrics like latency or energy efficiency.
* Extend the environment to model dynamic obstacles or moving agents.
* Experiment with different RL algorithms (e.g., DDPG, SAC) to compare performance.

**7. Contact Information**

For questions, feedback, or contributions, please reach out via the repository’s issue tracker or email the project maintainer at [saeidim@uow.edu.au].

Please cite our project if you use this code: