**Research and Development Report: Sector-Based Reinforcement Learning for Drone Navigation in Urban Environments**

**Abstract**

This research and development report outlines the design, implementation, and evaluation of a novel sector-based reinforcement learning (RL) framework for autonomous drone navigation in complex urban environments. By dividing the environment into interconnected sectors, the approach reduces state space complexity, enabling efficient and scalable navigation. Leveraging Unity's ML-Agents framework and the Proximal Policy Optimization (PPO) algorithm, the project achieves robust drone performance, with experimental results demonstrating high success rates, low collision rates, and optimized path efficiency in dense urban scenarios. The framework's implementation details, including source code and training configurations, are provided to support reproducibility and further research.

**1. Introduction**

**1.1 Background**

Urban environments pose significant challenges for autonomous unmanned aerial vehicles (UAVs) due to dense obstacle distributions, dynamic elements (e.g., moving vehicles), and the need for real-time decision-making. Traditional navigation methods, such as A\* or Rapidly-exploring Random Tree Star (RRT\*), struggle with computational scalability in such settings. Reinforcement learning offers a promising alternative by enabling adaptive, data-driven decision-making.

**1.2 Problem Statement**

The primary challenge is to develop a scalable and efficient navigation system for drones operating in complex urban environments, capable of avoiding obstacles, reaching designated targets, and optimizing flight paths under computational constraints.

**1.3 Objectives**

* Design a sector-based environment division strategy to reduce state space complexity.
* Implement a reinforcement learning framework using Unity ML-Agents and PPO for drone navigation.
* Evaluate the system's performance in simulated urban environments using metrics such as success rate, collision rate, path efficiency, and stability.
* Provide a reproducible framework with open-source code and detailed documentation.

**1.4 Significance**

This project advances autonomous UAV navigation by introducing a scalable, sector-based RL approach that balances local navigation tasks with global awareness. It has applications in urban delivery, surveillance, and disaster response, where efficient drone navigation is critical.

**2. Literature Review**

**2.1 Reinforcement Learning Overview**

Reinforcement learning (RL) involves an agent learning optimal actions through interaction with an environment, modeled as a Markov Decision Process (MDP). Key RL algorithms include:

* **Deep Q-Networks (DQN)**: Effective for discrete action spaces but less suitable for continuous control.
* **Proximal Policy Optimization (PPO)**: Balances sample efficiency and stability, ideal for continuous action spaces like drone control.
* **Deep Deterministic Policy Gradient (DDPG)**: Suitable for continuous control but prone to instability.

**2.2 Drone Navigation in Urban Environments**

Urban navigation requires robust sensor systems (e.g., LiDAR, RGB-D cameras) and path-planning algorithms. Challenges include obstacle density, dynamic objects, and limited computational resources onboard UAVs.

**2.3 Sector-Based Navigation**

Sector-based approaches, such as cell decomposition, partition environments into manageable regions, reducing computational complexity. Prior work includes grid-based navigation and hierarchical RL, but few studies integrate sector division with modern RL algorithms like PPO.

**2.4 Unity ML-Agents**

Unity ML-Agents provides a flexible platform for RL training in simulated environments. Its integration with Python and PyTorch supports scalable training, making it ideal for this project.

**2.5 Research Gaps**

* Limited exploration of sector-based RL for drone navigation.
* Lack of scalable frameworks balancing local and global navigation.
* Need for reproducible, open-source implementations.

**3. Methodology**

**3.1 System Overview**

The system comprises a Unity-based urban environment, a drone agent controlled via RL, and a sector-based navigation framework. The drone navigates to target locations while avoiding obstacles, guided by a PPO-trained policy.

**3.2 Urban Environment Simulation**

* **Setup**: A 3D urban environment in Unity with buildings, roads, and dynamic obstacles.
* **Tools**: Unity Editor (2021.3.22f1), Cinemachine for visualization, and custom assets for urban modeling.
* **Sector Division**: The environment is divided into interconnected sectors, each representing a localized navigation task (see part 2.pdf, Section 3.3).

**3.3 Reinforcement Learning Framework**

* **Algorithm**: PPO, selected for its stability and efficiency in continuous action spaces.
* **State Space**: Includes drone position, rotation, linear velocity, angular velocity, and normalized relative position to the target (see IP\_Drone\_Agent.cs, CollectObservations).
* **Action Space**: Continuous control over pitch, roll, yaw, and throttle.
* **Reward Function**: Combines rewards for distance reduction, orientation, alignment, and penalties for collisions, high velocity, and height deviations (see IP\_Drone\_Agent.cs, RewardParameters class).

**3.4 Training Process**

* **Curriculum Learning**: Progressively increases environment complexity (e.g., obstacle density, sector transitions).
* **Configuration**: Training parameters (e.g., learning rate, batch size) are defined in DroneNavigation.yaml (see part 2.pdf, Appendix A.1).
* **Hardware**: Training on CPU/GPU, with simulation speed optimized for real-time performance.

**3.5 Implementation Details**

* **Core Scripts**:
  + IP\_Drone\_Agent.cs: Manages RL logic, observations, actions, and rewards.
  + IP\_Drone\_Engine.cs: Simulates physics-based drone propulsion.
  + Normalization.cs: Normalizes observations using sigmoid and tanh functions.
  + TrailManager.cs: Visualizes drone paths for debugging.
* **Physics**: Rigidbody-based drone dynamics with mass and drag settings (see IP\_Base\_RigidBody.cs).
* **Input System**: Supports manual control for heuristic testing (see IP\_Drone\_Inputs.cs).

**4. Results**

**4.1 Training Performance**

* **Convergence**: Stable learning curves with cumulative reward growth (see part 2.pdf, Figure 4.1).
* **Curriculum Effectiveness**: Progressive training improved adaptability to complex scenarios.
* **Hyperparameter Impact**: Learning rate and reward multipliers significantly affected convergence speed.

**4.2 Navigation Performance**

* **Success Rate**: High success rates in reaching targets across various urban scenarios (see part 2.pdf, Table 4.1).
* **Collision Rate**: Low collision rates due to effective reward penalties.
* **Path Efficiency**: Sector-based navigation reduced path lengths compared to non-sector approaches (see part 2.pdf, Table 4.3).
* **Stability**: Minimal oscillations in drone orientation and velocity.

**4.3 Comparative Analysis**

* Outperformed baseline RL models (e.g., DDPG) and non-sector-based PPO in path efficiency and training time (see part 2.pdf, Section 4.4).
* Sector-based approach reduced state space complexity by up to 40%.

**5. Discussion**

**5.1 Key Strengths**

* Scalability through sector division.
* Robust performance in dense environments.
* Reproducible framework with open-source code.

**5.2 Limitations**

* Challenges in seamless sector transitions.
* Limited testing in real-world scenarios.
* Reward function tuning requires domain expertise.

**5.3 Implications**

* Applicable to urban delivery, surveillance, and search-and-rescue missions.
* Framework can be extended to other autonomous systems (e.g., ground robots).

**5.4 Future Enhancements**

* Integrate global path planning with local sector navigation.
* Explore dynamic sector resizing based on obstacle density.
* Validate with physical drones using vision-based sensors.

**6. Conclusion**

This project successfully developed a sector-based RL framework for drone navigation, demonstrating significant improvements in efficiency and scalability. The open-source implementation and detailed documentation provide a foundation for future research and real-world applications.