Continuous Maneuver Control And Data Capture Scheduling Of Autonomous Drone In Wireless Sensor Networks

Brendan Amoah-Daniels, Dylan Shattan, Bokre Samson

October 16, 2025

Ales Psaker

Abstract

This article addresses the issues of wireless drone sensor networks by using a new method that allows drones to adapt to new situations much quicker and more efficiently. This helps to reduce packet loss and improve network throughput. Recent advances in wireless sensing and autonomous drone technologies have made it possible to collect data effectively from large-scale, energy-harvesting ground sensor networks. They prove particularly useful in remote or disaster-susceptible areas, where the traditional communication infrastructure lacks reliability. The ground sensors are used to gather environmental data in the field and keep a queue to send to the drone. Energy constraints and the size of the buffer result in missing packets. The drone needs a painted battery with fixed power capacity. It can be programmed to move closer to the sensor for data copying using the dLOC policy. DDPG-MC combines this actor-critic network mechanism to offer real-time decision while balancing the flight dynamics of the drone and scheduling communication on collecting data from ground sensors. The quadcopter can only communicate with one of the sensors at any moment. It has a timestamp-based pressure of each sensors reading that TTA helps the drone in prioritizing on sensors that potential investigate time at risk.

Materials & Methods

This study used Python 3.5 and TensorFlow (an open-source machine learning library, used in this study for numerical computation) to simulate a drone flying over a square-kilometer field covered with several hundred sensors. Each sensor had a small battery and stored sensed data in a limited buffer. The drone was set to move around in such a way ('altitude hold mode') that it stayed at a constant altitude throughout its flight, while allowing it to adjust its heading and velocity when needed. The study implemented a mathematical idea, known as an absorbing Markov chain, to describe the drone's situation at any time. The absorbing Markov chain accounted for several factors, including the drone's battery life, the battery and memory levels of the ground sensors, the quality of the wireless signal between the drone and the sensors, and how long it had been since a sensor was last visited (called the 'Time-To-be-Alive,' or TTA, value). This model predicted how the system changed as the drone flew, interacted with sensors, or used energy. The simulation would stop when the drone's battery ran out, known as the 'absorbing state'.

Li et al. built an AI system called Deep Deterministic Policy Gradient-based Maneuver Control (or DDPG-MC) to operate on the drone while in the air. It is a type of deep reinforcement learning, meaning that the drone would learn through trial and error. Each flight would try different speeds, directions, and order-of-sensors to figure out which path is optimal, receiving a score (called the 'network cost') at the end to represent how much data got lost. Data from each 'episode' would then be uploaded into a 'replay memory' for further analysis. The system used an actor-critic network, which contained the actor (who suggested what actions the drone should take) and the critic (who judged how good the decision was based on the outcome). Over time, these components were meant to improve in tandem to better predict the best moves.

Analysis

Li et al.'s DDPG-MC algorithm offers a generic and efficient solution to the management of data collection within dynamic, multi-agent environments, which are concepts that can be easily adapted to ground-based autonomous systems. For instance, if it is a park scenario with multiple autonomous garbage-collecting vehicles that are moving around at the same time, the same reinforcement learning and trajectory planning methods can be leveraged to enhance the efficiency of operations as well as prevent collisions. The use of an interesting Markov chain model in the work provides environmental states and system alterations solid mathematical bases upon which to forecast and allows autonomous agents to make decisions not only on real-time sensory information, but on probabilistic descriptions of forthcoming conditions. This predictability is particularly important in active environments that are in a state of continuous change, like open public park areas in which population density and movement of human and animal populations are continually shifting.

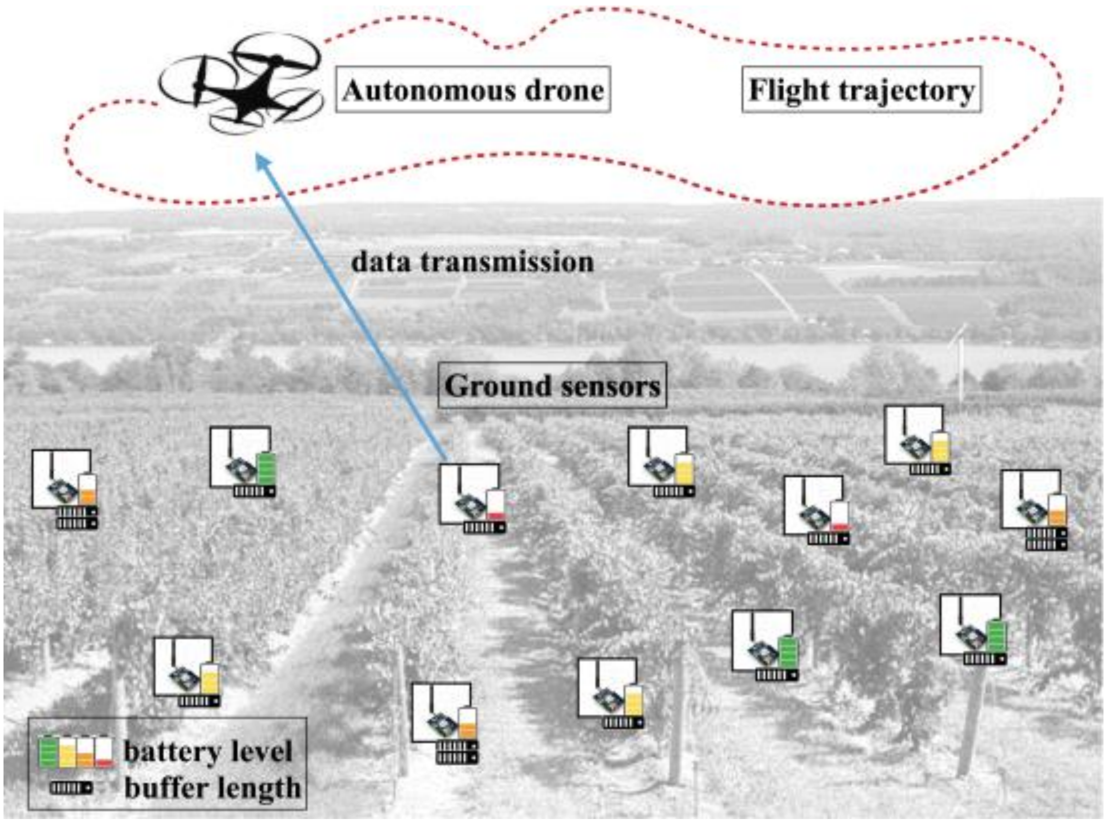

Figure 1: This depicts a drone being used to collect data from sensors in a vineyard relaying information about the ground conditions. The drone would use a DDPG-MC algorithm to predict the best path and velocity to stay within range of the sensors to optimize energy consumption, time, and data loss.

The actor-critic network of the DDPG-MC model can be reused to control navigation and task scheduling of waste-collecting robots. For example, the "actor" module would choose energy-saving paths through the park to reach high-litter areas with minimal energy expenditure, and the "critic" would estimate how energy-efficient these paths are in terms of battery use, time, and collection efficiency. The same closed-loop learning process the drone utilizes can be applied to help the ground robots continually improve performance on subsequent cleaning runs even while park layouts, ground conditions, or traffic conditions evolve over time. Moreover, the study focus on communication burden-motion dynamics tradeoff is aligned with that of collaborative cleaning vehicles. Just like the drone could only talk to one ground sensor at a time, each robot will be able to talk to only a few closer units or central servers at any given moment. Implementing a variant of the DDPG-MC scheduling algorithm will help coordinate communication among a cluster of robots so that trash density, filled bins, and coverage route data get shared without network overload and redundant overlapping.

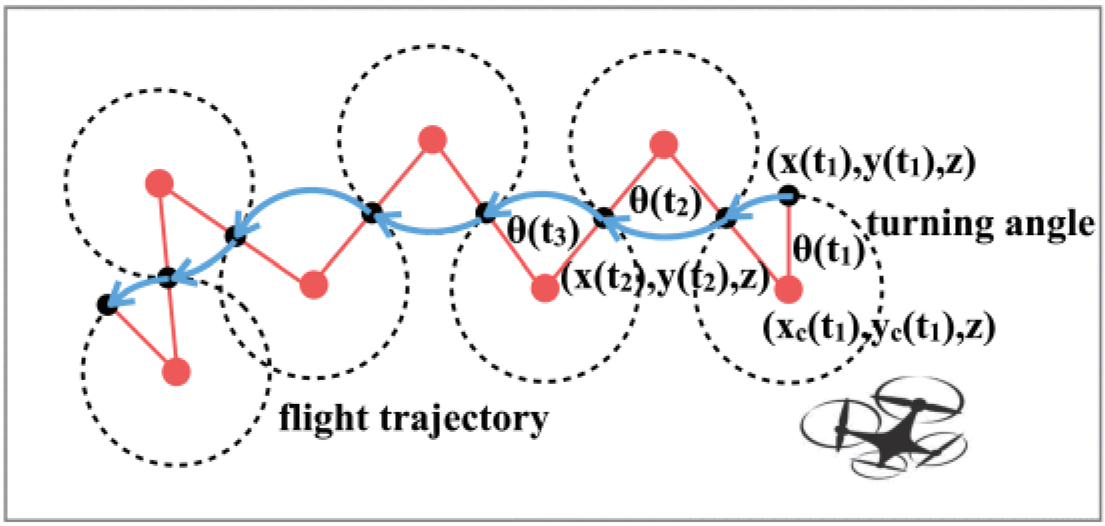

Figure 2: This depicts an optimized path created by a DDPG-MC framework, calculated to spend the needed amount of time within range of a sensor to collect and send data, and spend the least amount of time out of range of a sensor in order to transfer data as quickly as possible.

By implementing Li et al.'s aerial data-acquisition model on ground-based garbage collection networks, robot programmers are able to self-adaptively re-prioritize actions should unexpected obstacles—such as pedestrians, cyclists, or wildlife—arise, much as the drone re-directed course when approaching sensors that had time-critical Time-To-be-Alive values. By such adaptive decision-making, park-cleaning robots might maintain efficient operation, minimize energy wastage, and avoid disruption of the populace. Together, the DDPG-MC approach offers a fascinating framework for achieving intelligent, adaptive coordination in commonly used public areas where autonomy, security, and communication efficiency are all critical.

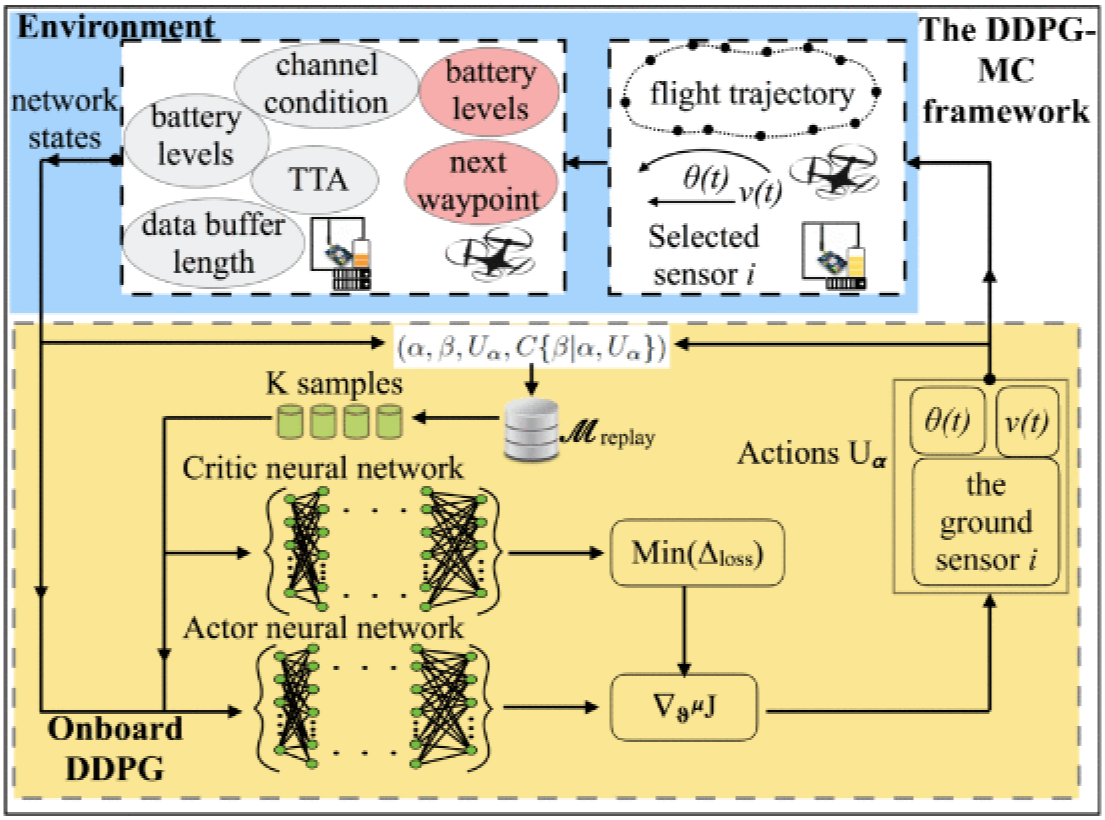

Figure 3: This is a detailed drawing of DDPG-MC framework, showing all the factors taken into account by the program while predicting and reinforcing optimal flight paths.

While creating a drone to clean up trails or fields at parks, we could use DDPG-MC algorithm to optimize flight paths from different points of interest, such as trails and fields that are commonly used. In addition to optimizing flight paths, the sensors could be used to measure the amount of people and the length of time they spend there. Logically, the more people pass through an area, and the longer they spend there, there will be a higher chance of there being trash there. The ground sensors could be used to predict the amount of trash in an area, making it a higher priority foe being cleaned, and if the sensors detect that there are people there, then it could avoid those areas until they leave.

References

Li, K., Ni, W., & Dressler, F. (2021). Continuous Maneuver Control and Data Capture Scheduling of Autonomous Drone in Wireless Sensor Networks. *IEEE Transactions on Mobile Computing*, 1–1. https://doi.org/10.1109/tmc.2021.3049178