

Article

Residual Learning for Riemannian Motion Policies in Multi-Drone Obstacle Avoidance and Communication-Aware Navigation

Liping Shi , Esben Haubro Skov , Asbjørn Lybker Christensen and Rune Hylsberg Jacobsen * 

Department of Electrical and Computer Engineering, Aarhus University, 8200 Aarhus N, Denmark

* Correspondence: rhj@ece.au.dk

Abstract

Autonomous multi-drone navigation in unknown environments requires robust solutions for obstacle avoidance, formation cohesion, and inter-drone communication maintenance. We address these challenges using Riemannian Motion Policies (RMPs), a second-order dynamic framework for motion generation, within the RMPflow architecture. We design handcrafted and learned obstacle avoidance policies, including circle-based and LiDAR-based approaches, and introduce residual reinforcement learning as a novel mechanism for refining suboptimal handcrafted policies. To maintain reliable drone-to-drone communication, we propose a radio link quality policy that dynamically adapts the maximum permitted inter-drone distance based on signal strength, contrasting with a fixed worst-case baseline. These policies are composed within RMPflow into an efficient reactive motion planner and evaluated through 2D simulations across multiple configurations. Results show that residual learning substantially improves a suboptimal handcrafted policy, reducing collisions by 35.4% in drone formations and up to 74.7% for single drones. Notably, a policy trained without any initial obstacle avoidance capability failed to obtain such improvement, underscoring the importance of informed policy initialisation. The adaptive radio link policy achieved 20% to 25% shorter navigation time through randomly generated environments compared to a fixed maximum distance constraint, demonstrating the practical value of radio signal-aware formation control.

Keywords: drone; UAV; motion planning; Riemannian Motion Policy; reinforcement learning

1. Introduction

Autonomous drones have the potential to enhance efficiency, safety, and speed in civil infrastructure inspection [1]. Completing such missions quickly and efficiently motivates the use of multiple collaborating drones, which can inspect different parts of an infrastructure simultaneously, reducing overall mission time. Effective collaboration requires drones to communicate, coordinate task allocation [2], and avoid collisions with both obstacles and one another while planning paths toward their assigned goals [3,4]. Operating in unknown environments further demands a motion planning algorithm capable of generating collision-free trajectories at both global and local levels. This requires short-term reactive strategies, such as obstacle avoidance, alongside long-term path planning to handle dynamic changes and enable free movement through unstructured environments [5].

Motion planning algorithms can be organised into three broad categories: classical planning, reactive policy-based planning, and learning-based planning. Classical



Academic Editor: DaeEun Kim

Received: 30 April 2026

Revised: 20 June 2026

Accepted: 22 June 2026

Published: 30 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

approaches, such as A* [5] and Rapidly-exploring Random Trees (RRT) [6] search over discretised or sampled configuration spaces to find collision-free paths and are well-suited to structured environments with known maps. Optimisation-based approaches, such as Model Predictive Control (MPC) [7], generate safe and efficient trajectories by solving constrained optimisation problems that account for vehicle dynamics and mission objectives, but typically require considerable onboard computation and prior environment knowledge. Reactive policy-based approaches, including potential field methods [8], the Dynamic Window Approach (DWA) [9], and Optimal Reciprocal Collision Avoidance (ORCA) [10], encode motion generation as structured local policies that respond directly to the drone's current state and sensory observations.

Riemannian Motion Policies (RMPs) [11] provide a more mathematically rigorous instantiation of this idea, representing each motion objective, such as goal attraction or obstacle repulsion, as a policy combining a desired acceleration with a metric tensor that encodes the local importance of that behaviour. The RMPflow framework [12] then provides a principled mechanism for composing multiple RMPs defined across different task spaces into a single unified motion policy, resolving conflicts between competing objectives in a geometrically consistent manner. These policy-based methods offer low computational requirements, rapid response times, and natural robustness to uncertainty, making them well-suited to dynamic and unknown environments. Learning-based approaches, by contrast, train models, often using reinforcement learning or imitation learning, to acquire navigation and control behaviours from experience [13].

In this paper, we study the case of drones collaborating in an unknown environment, where individual drones may have conflicting objectives arising from their spatial relationships to goals and to one another. A further complication is that the effective communication range between drones varies with environmental conditions, making it necessary to dynamically manage inter-drone distances to prevent drones from becoming disconnected from the formation. We address these challenges by bridging the reactive policy-based and learning-based paradigms within the RMPflow framework [12], a system for composing collections of RMPs into a unified motion policy, operating on the drone's current state and sensory observations. Specifically, we investigate whether a suboptimal handcrafted obstacle avoidance policy can be refined through residual reinforcement learning, and whether communication link quality can be used to adaptively govern formation constraints. Thereby, we combine the mathematical transparency of RMPs with the adaptive power of learned behaviour. The main contributions of this work are as follows.

- We introduce a residual reinforcement learning approach to supplement and improve a handcrafted RMP obstacle avoidance policy.
- We design a radio link quality policy that dynamically adapts the maximum inter-drone distance based on communication signal strength, improving navigation efficiency over fixed worst-case constraints.
- We compose these policies within RMPflow into a unified reactive motion planner and validate its performance through 2D simulations across multiple configurations.

Our approach introduces a novel use of residual learning for RMP generation in a multi-drone navigation setting operating in unknown environments, enabling the first comparative evaluation of distinct learning-based methods with different objectives in this context. Unlike prior work applying residual learning within RMP frameworks, which has been demonstrated exclusively for single-robot manipulators in controlled settings, this work advances understanding of RMP adaptability and effectiveness in dynamic, multi-drone environments. The approach is further distinguished by a communication-aware formation policy that dynamically adapts inter-drone distance constraints based on radio signal strength.

The rest of the paper is organised as follows. Section 2 reviews related work on RMP-based motion planning, residual learning, and multi-UAV coordination. In Section 3, we present our design of policies for the motion planner. We describe our system implementation in Section 4 and present results from our simulation studies in Section 5. A discussion and conclusion follow in Sections 6 and 7, respectively.

2. Related Work

RMPs belong to a class of reactive motion policies that parametrise non-Euclidean behaviours as dynamical systems in intrinsically non-linear task spaces [11]. Cheng et al. [12] proposed RMPflow to synthesise local control policies into an expressive global policy. RMPflow accomplishes this by integrating multiple RMPs to create a versatile global motion policy while exploiting sparse structures to enhance computational efficiency. RMPflow uses a tree structure to calculate and relate local policies to the global configuration space of a robot. The authors defined a class of motion policies called Geometric Dynamical Systems (GDSs) [14]. These are procedures for defining policies based on GDSs that come with a stability guarantee. The research presented example policies and demonstrates that the GDS-based RMP within the RMPflow tree structure can effectively control a dual-arm robotic manipulator in environments containing obstacles, through the implementation of goal-reaching and obstacle avoidance policies. More recently, geometric fabrics [15] have been proposed as a generalisation of RMPs that add provable stability guarantees while retaining the modular design principles that make RMPs effective.

Residual reinforcement learning, which learns a corrective signal on top of an existing controller, has shown promising results in robot control settings [16,17]. Aljalbout et al. [18] proposed a method for learning vision-based reactive obstacle policies using the RMPflow framework. The objective is to learn the relationship between visual inputs and motion representations to avoid obstacles for a robot manipulator. Reinforcement learning enables the production of an optimised policy compared to defining a handcrafted policy. The learned policy is formulated as an RMP, which allows for efficiently combining it with other RMP policies in the RMPflow framework. Two different approaches to learning an RMP policies are evaluated. The first approach trains drones to learn both to reach a goal position and avoid obstacles. The second approach trains only obstacle avoidance via Residual Policy Learning (RPL) with a handcrafted RMP goal attractor policy. RPL involves learning a corrective adjustment to an existing base policy, allowing for fine-tuning and adaptation to specific tasks without fully retraining the original policy. This latter approach proved to be more efficient during training. Tests in an environment with one obstacle show that the second approach achieved more than twice the success rate in reaching the goal position without collisions compared to the first approach.

Li et al. [19] proposed a more efficient message-passing algorithm compared to RMPflow, called RMP². This algorithm relies on auto-differentiation, which allows for differentiation through the RMP tree for better integration with machine learning algorithms. The authors compared reinforcement learning training results of three types of policies within the RMP² framework. The first policy is learned as a randomly initialised neural network in an end-to-end reinforcement learning fashion. The second policy uses a residual neural network architecture applied to a handcrafted RMP² policy consisting of multiple RMP policies for goal reaching, joint damping, joint velocity limits, and collision avoidance. The third policy is residual to a handcrafted goal-reaching policy, and its output is concatenated before combining it with the same handcrafted RMP² used in the second policy. Training results generally show a higher collision rate for the first policy than for the residual approaches, which achieve low collision rates. The authors conclude that high rewards during training do not necessarily lead to desired behaviours, demonstrating that

the design of good reward functions is non-trivial [19]. Both Aljalbout et al. [18] and Li et al. [19] achieve promising results using reinforcement learning in a residual architecture with handcrafted RMP policies, though both research groups focus on robot manipulators in controlled settings. Li et al. [19] rely on prior knowledge about the environment, whereas Aljalbout et al. [18] rely on a vision setup to detect obstacles.

RMPs for navigating a remote control car in environments with obstacles were studied by Meng et al. [20]. The authors compared an expert model consisting of handcrafted obstacle avoidance and goal attraction policies to three system variations using neural networks. The obstacle avoidance policy uses a discrete set of points abstractly representing the environment, obtained using either a laser scanner or a neural network trained to estimate distances from visual input. Using a neural network trained to predict an RMP based on visual information and the state of the remote control car yielded results that were occasionally better than the handcrafted expert policies.

Pantic et al. [21] proposed a method for using RMPs directly on volumetric maps and LiDAR scans for obstacle avoidance on Micro Aerial Vehicles (MAVs). Rather than relying on smoothed or pre-processed map representations, their approach used GPU raycasting to evaluate large numbers of concurrent RMP policies directly on voxelised maps and raw LiDAR data, achieving planning rates in the kilohertz range. The modular nature of RMPs facilitates straightforward integration of new policies, and the method was demonstrated successfully on a real MAV in both static and dynamic obstacle environments. The work by Pantic et al. is closely related to the LiDAR-based obstacle avoidance policy developed in the present work, as both exploit raw range measurements to define RMP policies directly, without requiring an intermediate map representation. However, whereas Pantic et al. focus on maximising planning throughput on a single MAV using GPU acceleration, our approach targets multi-drone coordination in unknown environments and is designed to be combined with formation cohesion and communication maintenance policies within the RMPflow framework. The feasibility of RMP-based reactive planning on real aerial robots has been further demonstrated by Shi et al. [22], who apply Forced Geometric Fabrics to rope manipulation and collision avoidance in real-world spray painting experiments.

Liu and Bucknall [23] compared different methods for formation control, concluding that behaviour-based formation control is more effective in handling multitask missions than leader–follower or virtual structure control schemes, while also being easier to adapt to different conditions, though potentially requiring more communication between drones. The authors noted that hybrid solutions combining different control strategies may be preferable. Building on this, Li et al. [24] explored how RMPs may implement formation control, designing policies for centralised and decentralised control of multiple mobile robots with a focus on behaviour-based formation control. A hybrid behaviour and leader–follower-based formation control scheme was implemented, with collision avoidance and formation maintenance policies on each robot and navigation toward a goal implemented only for the leader robot. The authors achieved stable formations with a fixed structure while avoiding collisions. In [25], Li et al. simulated a drone swarm with the requirement that drones preserve connectivity within a predefined radius of other drones, successfully navigating cluttered environments. Meng et al. [26] complement these RMP-based approaches by introducing a lightweight, reinforcement learning-based real-time path planning method for drones, optimising network architecture using cross-layer connections to improve feature retention and generalisation, achieving rapid convergence and collision-free paths with strong robustness to dynamic environments.

Optimisation-based approaches have been widely investigated for multi-UAV coordination [7,27]. These methods generate safe and efficient trajectories by solving constrained optimisation problems that account for UAV dynamics, inter-UAV collision

avoidance, and mission objectives. For instance, Richards and How [7] formulated the multi-vehicle collision avoidance as a mixed integer linear programme, while Shi et al. [4] modelled multi-UAV coordination as a Generalised Potential Game solved iteratively in a decentralised manner. Distributed model predictive control has also been widely applied to UAV swarm trajectory planning, though it requires considerable onboard computation that increases weight and power consumption [28]. These computational demands make optimisation-based methods generally less suited to unknown and dynamic environments requiring rapid reactive responses, motivating the use of reactive policy-based approaches such as RMPs in the present work.

The prior work reviewed above establishes a strong foundation for RMP-based robot navigation in obstacle-constrained environments. However, two important gaps remain. First, residual learning approaches for RMP generation have been demonstrated primarily for robot manipulators in controlled, single-robot settings, with limited exploration of their applicability to multi-drone navigation in unknown environments. Second, communication-aware formation control within the RMP framework has received little attention. The present work addresses both gaps by extending RMP-based planning to multi-drone navigation in unknown environments. Specifically, we introduce a circle-based and a LiDAR-based obstacle avoidance policy, refine the latter through residual reinforcement learning, and propose a radio link quality policy that dynamically adapts formation constraints based on communication signal strength.

3. Motion Planning Policy Design

RMPflow provides the framework for policy integration [12], allowing a complex robotic control problem to be decomposed into simpler subtasks, each defined on its own manifold and described by an RMP [29]. Subtasks and their RMPs can be individually designed and tested due to the linearity properties of RMPs. Once defined, RMPflow consistently combines RMPs to produce a global policy for the entire robotic system.

The proposed motion planner links RMPs to generate the motion of the drones in the environment (Figure 1). It involves three subtasks: goal attraction (ga), obstacle avoidance (oa), and keeping a maximum distance (md) between drones. For each subtask, we design or learn its RMPs to define the policy acceleration for the drone (dr). Finally, we integrate the acceleration to obtain the trajectories for the drones to follow.

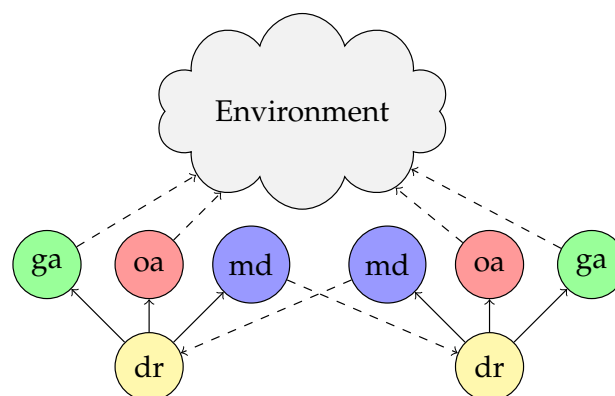


Figure 1. The example illustrates the configuration of motion policies for two drones. Solid and dashed arrows symbolise the hierarchical relationship between policies and information flow, respectively.

3.1. Goal Attractor Policy

The goal attractor policy follows the design of [14,24]. The policy exerts a policy acceleration for the drone to move toward a goal position. The policy is based on a GDS that can be written as [14]:

$$\ddot{\mathbf{x}}^d = -\mathbf{M}_{ga}(\mathbf{x})^{-1} \left(\nabla_{\mathbf{x}} \Phi_{ga}(\mathbf{x}) + \zeta_{\mathbf{M}_{ga}}(\mathbf{x}, \dot{\mathbf{x}}) + \mathbf{B}_{ga}(\mathbf{x}, \dot{\mathbf{x}}) \dot{\mathbf{x}} \right), \quad (1)$$

where $\ddot{\mathbf{x}}^d$ is the desired acceleration, $\mathbf{M}_{ga}(\mathbf{x})^{-1}$ is the inverse priority matrix representing the importance of the policy when integrating with other policies, $\nabla_{\mathbf{x}} \Phi_{ga}(\mathbf{x})$ is the gradient of a potential function defining the change of acceleration relative to the distance to the goal, $\zeta_{\mathbf{M}_{ga}}(\mathbf{x}, \dot{\mathbf{x}})$ is a curvature term associated with the priority matrix $\mathbf{M}_{ga}$ to maintain path consistency across different traversal speeds [24], $\mathbf{B}_{ga}(\mathbf{x}, \dot{\mathbf{x}})$ is a damping term to stop the motion when the goal is reached and provide stability, and $\dot{\mathbf{x}}$ is the velocity of the drone.

3.2. Obstacle Avoidance Policy

We examine four different approaches for drone obstacle avoidance. The first approach uses a circle-based obstacle avoidance policy [24], which ensures a safety distance around a point such that an obstacle may be interpreted as a point with a safety distance acting as a barrier. The second approach derives its RMP from the circle-based policy in [24] but removes the assumption of obstacles as known points with a fixed safety distance. Instead, it uses LiDAR to place points with repulsive policy accelerations at the obstacle surface. The LiDAR-based obstacle avoidance policy does not assume prior knowledge of the environment, as observation of the environment is performed as part of the policy evaluation, and can handle arbitrarily shaped obstacles in contrast to the circle-based approach. The third and fourth approaches apply residual policy learning using a neural network together with LiDAR input in different configurations, combined with a handcrafted LiDAR-based obstacle avoidance policy and goal attractor policy.

In scenarios with multiple drones, each drone additionally uses an inter-drone collision avoidance policy, which is a variant of the circle-based obstacle avoidance policy in which each drone is treated as an obstacle whose position is updated at each simulation step.

3.2.1. Circle-Based Obstacle Avoidance

The circle-based obstacle avoidance policy is inspired by [24] and follows the design approach of [14]. It ensures that the drone does not collide with a given obstacle, and multiple instances of the policy are composed within RMPflow when several obstacles are present in the drone's vicinity. The policy operates in a one-dimensional task space defined by [24]:

$$z = \psi(\mathbf{x}) = \frac{\|\mathbf{x} - \mathbf{c}\|}{d_s} - 1, \quad (2)$$

where $\psi(\mathbf{x})$ is the task map that transforms the drone position from the configuration space to the one-dimensional task space, and $z \in \mathbb{R}$ is the resulting task space coordinate formed as the Euclidean distance between the drone position $\mathbf{x}$ and the centre of the obstacle $\mathbf{c}$, normalised by a safety distance d_s . The safety distance is achieved by scaling the task space, with the effect that the policy output increases more rapidly when the safety distance is small. The policy acceleration $\mathbf{f}$ of the corresponding RMP is given by:

$$\mathbf{f} = -\mathbf{M}_{oa}(z)^{-1} \left(\nabla_z \Phi_{oa}(z) - \zeta_{\mathbf{M}_{oa}}(z, \dot{z}) - \mathbf{B}_{oa}(z, \dot{z}) \dot{z} \right), \quad (3)$$

where $\dot{z}$ is the rate of change of the normalised distance between the drone and the obstacle centre. The term $\zeta_{\mathbf{M}_{oa}}(z, \dot{z})$ is a curvature term associated with the priority matrix $\mathbf{M}_{oa}(z, \dot{z})$,

maintaining geometric consistency of the policy across different traversal speeds. The gradient of the potential function is [24]:

$$\nabla_z \Phi_{oa}(z) = \alpha_{oa} w_{oa}(z) \nabla w_{oa}(z) . \quad (4)$$

α_{oa} is a user-definable scalar damping coefficient that governs the steepness of the potential function, and $w_{oa}(z)$ is a weight factor. The priority matrix $\mathbf{M}_{oa}(z)$ is given by [14,24]:

$$\mathbf{M}_{oa}(z) = \mathbf{G}_{oa}(z, \dot{z}) + \frac{1}{2} \dot{z} w_{oa}(z) \nabla u(\dot{z}) \mathbf{I} , \quad (5)$$

where $\mathbf{I}$ is the identity matrix. The metric $\mathbf{G}_{oa}(z, \dot{z})$ is defined as [24]:

$$\mathbf{G}_{oa}(z, \dot{z}) = w_{oa}(z) u(\dot{z}) \mathbf{I} , \quad (6)$$

where

$$w_{oa}(z) = \frac{1}{z^4} \quad (7)$$

ensures that the circle-based obstacle avoidance policy has a large priority when the drone is close to an obstacle [24]. The $u(\dot{z})$ term introduces a dependency on the velocity of the drone that raises the priority of obstacle avoidance when the drone moves toward the obstacle and is reduced when the velocity vector $\dot{\mathbf{x}}$ points away from it. $u(\dot{z})$ is expressed as:

$$u(\dot{z}) = \epsilon_{oa} + \min(0, \dot{z}) \dot{z} , \quad (8)$$

where ϵ_{oa} is a small positive number that acts as a regularisation term that keeps the policy numerically stable by ensuring the metric is always positive definite, even when the drone is moving away from the obstacle. Li et al. [24] define the potential function for the obstacle avoidance policy as:

$$\Phi_{oa}(z) = \frac{1}{2} \alpha_{oa} w_{oa}(z)^2 . \quad (9)$$

This potential function provides a repulsive policy acceleration when the drone is close to the obstacle, increasing asymptotically toward infinity as z approaches zero. Finally, the damping term is defined as [24]:

$$\mathbf{B}_{oa}(z, \dot{z}) = \eta_{oa} \mathbf{G}_{oa}(z, \dot{z}) , \quad (10)$$

where η_{oa} is a positive scalar.

3.2.2. LiDAR-Based Obstacle Avoidance Policy

Figure 2a illustrates the LiDAR sensor model, in which each ray is associated with a dedicated one-dimensional obstacle avoidance policy. These policies are connected as children of a LiDAR obstacle avoidance node in an RMP tree [24], as shown in Figure 2b.

Since the LiDAR places obstacle avoidance policy centres on the surface of obstacles, the safety distance does not relate to the size of the obstacle. The safety distance should therefore be reduced to avoid excessive safety margins that would hinder drones from navigating cluttered environments. An instability issue has been observed in simulations using the circle-based obstacle avoidance policy from [24] when the RMP output is sampled at a low rate. If the drone velocity is sufficiently high relative to the safety distance, the drone may traverse a significant portion of the safety margin between two consecutive time steps, causing the RMP to output an unrealistically large policy acceleration. This issue is also likely to arise when the LiDAR resolution is insufficient to detect an obstacle promptly, potentially leading to an unstable system or unrealistic accelerations. The effect is more

pronounced for small obstacles than large ones, because the task map $\psi(\mathbf{x})$ in Equation (2) scales the task space according to the safety distance. To address this issue, we define an alternative task map $\psi(\mathbf{x})$ where the one-dimensional task space is formed simply by the direct Euclidean distance between the drone and the obstacle:

$$z = \psi(\mathbf{x}) = \|\mathbf{x} - \mathbf{c}\|, \quad (11)$$

where $\mathbf{x}$ is the drone position and $\mathbf{c}$ denotes the contact point between the LiDAR ray and the obstacle. This task map does not enforce a safety distance and does not guarantee collision avoidance in all cases. However, collisions are unlikely when the α_{oa} parameter of the potential function (cf. Equation (4)) is set sufficiently high.

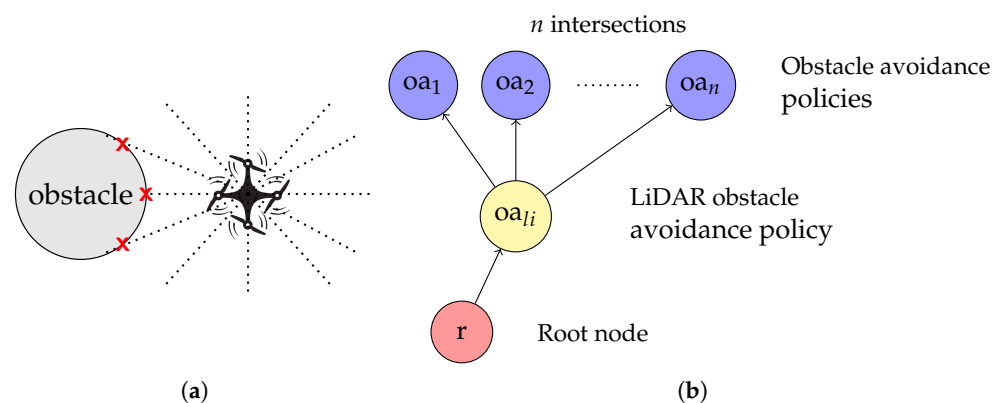


Figure 2. (a) LiDAR-based obstacle avoidance policies using points with a safety radius. (b) RMP tree for LiDAR-based obstacle avoidance. The drone icon is attributed to [vecteezy.com](https://www.vecteezy.com) (accessed on 3 April 2026).

3.3. Learned Policies for Obstacle Avoidance

We present two compositions of residual policies for obstacle avoidance, shown in Figures 3 and 4. RPL, introduced by [17], uses deep reinforcement learning to learn a residual correction signal on top of an initial controller. The residual policies are parametrised as actor networks composed of fully connected neural network layers, consisting of an input layer, two hidden layers of sizes 128 and 64, and an output layer of size five. We define $\mathbf{o}$ as a vector of observed LiDAR ranges. In our study, we use 20 uniformly spaced rays. The input layer has size 22 ($\dot{\mathbf{x}}, \mathbf{o}$) for the first composition and size 28 ($\dot{\mathbf{x}}, \mathbf{f}, \mathbf{M}, \mathbf{o}$) for the second, where ($\mathbf{f}, \mathbf{M}$) describes the desired policy acceleration and priority matrix from the LiDAR-based obstacle avoidance policy, $\dot{\mathbf{x}}$ contains the drone velocity, and $\mathbf{o}$ contains the observed LiDAR ranges.

The observation space defines the set of environment states available to the policy, and the action space defines the set of outputs the policy can produce. Actions are defined as a concatenation of the motion policy output. To ensure stability of the learned system, the priority matrix $\mathbf{M}$ must be positive definite [30]. The priority matrix is therefore parametrised by the Cholesky decomposition $\mathbf{M} = \mathbf{L}\mathbf{L}^T$, where $\mathbf{L}$ is a lower triangular matrix with positive diagonal entries [30]. When combining a residual policy with a handcrafted RMP policy, the policy accelerations and priority matrices are added. The combined priority matrix $\mathbf{M}_{comb}$ is defined as [19,30]:

$$\mathbf{M}_{comb} = (\mathbf{L}_{res} + \mathbf{L}_{hc})(\mathbf{L}_{res} + \mathbf{L}_{hc})^T, \quad (12)$$

where $\mathbf{L}_{res}$ and $\mathbf{L}_{hc}$ emerge from the residual and handcrafted policies, respectively. The combined policy acceleration and priority matrix is applied within the RMPflow framework after combination.

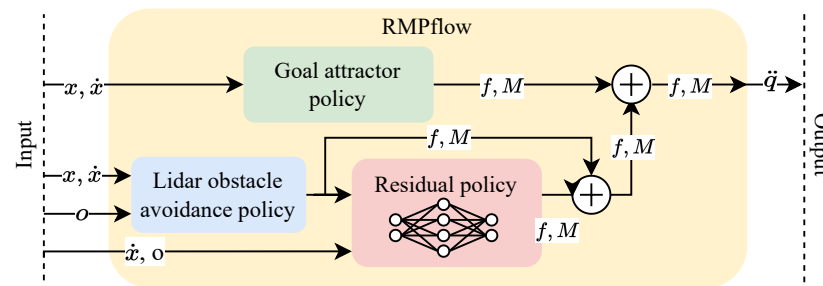


Figure 3. A sketch of the LiDAR-based residual RMP design for obstacle avoidance.

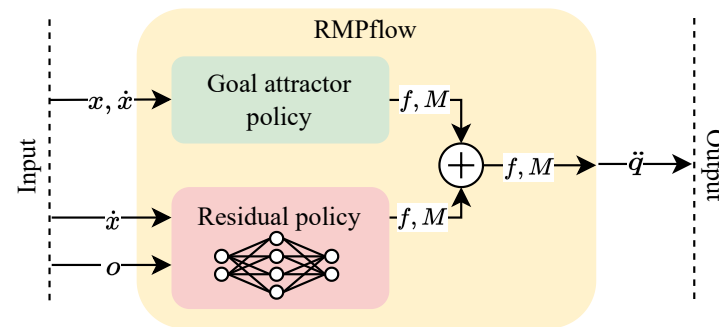


Figure 4. A sketch of the LiDAR-based residual acceleration design, inspired by [18,19].

3.3.1. LiDAR-Based Residual RMP

Figure 3 presents the LiDAR-based residual RMP design, inspired by the residual learning building block from the ResNet architecture [31]. It consists of a LiDAR-based obstacle avoidance module using a handcrafted RMP, a residual policy module using a neural network trained via deep reinforcement learning, and a goal attractor policy module using a handcrafted RMP. The handcrafted LiDAR-based policy provides a reliable baseline, enabling the residual network to focus on correcting limitations such as oscillations rather than learning obstacle avoidance from scratch. This ensures efficient integration of learning-based enhancements while retaining the robustness of the baseline policy.

The second composition, named LiDAR-based residual acceleration and illustrated in Figure 4, is inspired by [18,19]. In contrast to the first composition, it uses a learned residual policy for obstacle avoidance without any handcrafted RMP policy module. In Figures 3 and 4, the dashed lines abstract the inputs and outputs in the RMPflow tree $(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{q}})$. The position and velocity of the drone in the task space are represented by $\mathbf{x}$ and $\dot{\mathbf{x}}$, respectively, $(\mathbf{f}, \mathbf{M})$ refers to the policy output, and $\ddot{\mathbf{q}}$ refers to the integrated acceleration for the drone from all task spaces and $\mathbf{o}$ is the LiDAR ray ranges.

3.3.2. Reward Function

The reward function guides the training of residual policies by evaluating the desirability of actions at each time step. The total reward is a weighted sum of six components: reaching the goal, avoiding collisions, minimising time, reducing travel distance, controlling accelerations, and limiting unnecessary traversal. Rewards for goal reaching and penalties for collisions are applied only at the end of each episode, emphasising these as primary outcomes. Continuous components incentivise quicker progress and efficient paths, while control penalties discourage large accelerations.

Component weights are calibrated to prioritise collision avoidance, goal reaching, travel distance minimisation, and time efficiency, with lesser emphasis on acceleration control. Calibration is based on estimated cumulative rewards in representative scenarios, assuming 200 steps to reach the goal under typical conditions. For instance, under this assumption, the cumulative goal-reaching reward amounts to 15, which is outweighed

by the collision penalty of 20, reflecting the primacy of safety over navigation efficiency. Rewards are bound to ensure training stability.

The 200-step estimate used for weight calibration was derived from preliminary simulations of the training environment without a trained policy. Since the time and control penalty terms are continuous per-step components, longer episodes naturally accumulate proportionally larger penalties, preserving the incentive for efficient navigation regardless of actual episode length. The 200-step figure therefore serves as a calibration reference for the relative scaling of reward components rather than a constraint on episode behaviour.

The reward function configuration presented above was selected based on a shaping study in which five configurations were evaluated, progressively adding time, control, and traversed distance penalty terms on top of a base reward comprising goal-reaching, collision, and distance components. The full combination was found to produce the best navigation behaviour in the training environment.

3.4. Maximum Distance Policies

A maximum distance policy controls the separation between drones, acting as a virtual leash that pulls one drone toward another when their distance approaches a maximum allowed value d_{max} (Figure 5). Two policies are designed: one that keeps the maximum distance at a fixed preset value, and one that determines the maximum distance dynamically to maintain communication quality based on a radio communication model. The policy outputs a desired policy acceleration $\mathbf{f}$ that increases as the distance between drones approaches d_{max} .

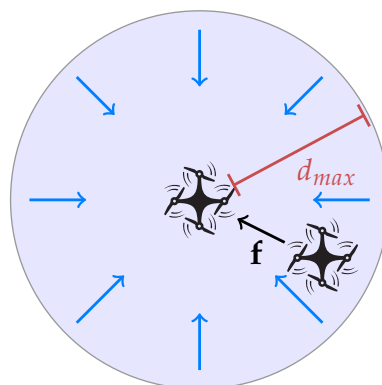


Figure 5. Illustration of the maximum distance policy.

The motion planner uses two maximum distance variants: the fixed maximum distance and the radio-based maximum distance. The fixed maximum distance policy keeps each drone within a radius of the target drone. The task map $\psi(\mathbf{x})$ to a one-dimensional task space is formed based on the drone to target distance:

$$z = \psi(\mathbf{x}) = \left(\frac{\|\mathbf{x} - \mathbf{c}\|}{d_{max}} \right) - 1 \quad (13)$$

where $\mathbf{x}$ and $\mathbf{c}$ denote the positions of the drone and the target, respectively, and d_{max} is the desired maximum distance. The policy acceleration is designed using a hyperbolic tangent function to bound the output with user-definable parameters:

$$\mathbf{f} = \tanh\left(\frac{-\alpha_{md} \exp(z \beta_{md})}{\gamma_{md}}\right) \gamma_{md}, \quad (14)$$

where α_{md} shifts the curve of the policy acceleration, β_{md} controls its steepness, and γ_{md} sets the maximum magnitude. The priority matrix is defined as the policy weight $\mathbf{M} = w_{md}\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Figure 6 illustrates the effects of these parameters.

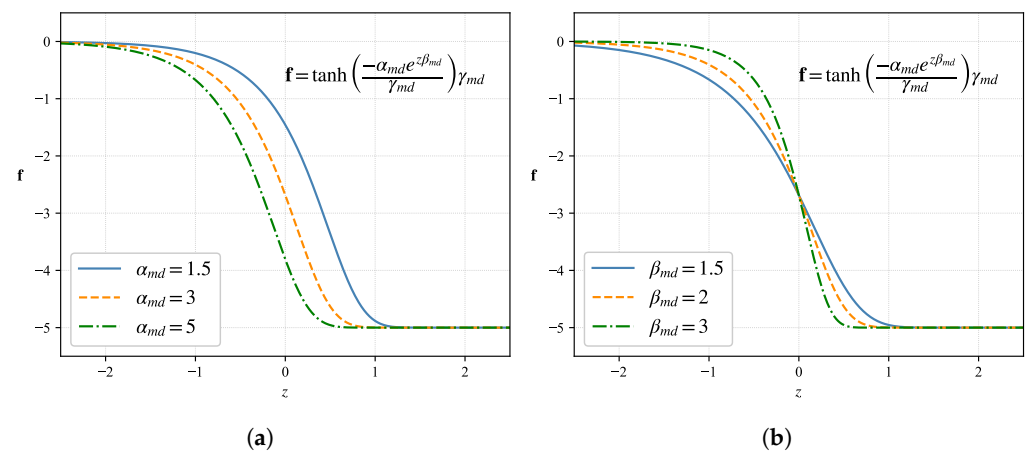


Figure 6. Parameters of the policy acceleration function for the fixed maximum distance policy. In both plots, $\gamma_{md} = 5$ was used. (a) β_{md} is kept constant while α_{md} is varied. (b) α_{md} is kept constant while β_{md} is varied.

The radio-based maximum distance policy estimates the maximum allowable distance between drones to maintain a specified signal quality, quantified by the Received Signal Strength Indicator (RSSI) of the radio communication. The maximum distance d_{max} is varied gradually and proportionally in response to the signal quality margin, without relying on a radio propagation model. We acknowledge the potential for refining the modelling of path loss between drones in the group to provide a more accurate representation of RSSI variations. However, due to the complexity and dynamic nature of both the group of drones and the environment, such studies often rely heavily on empirical investigation [32].

The maximum distance update has two modes, as shown in Equation (15). In the first mode, the update is proportional to the current signal quality margin $RSSI_{\Delta}(t) = RSSI(t) - RSSI_{desired}$, which is the difference between the current signal quality $RSSI(t)$ and the desired signal quality $RSSI_{desired}$. This mode is active when the drone distance d_{Δ} is within a threshold distance d_{thres} of the current maximum distance, or when the signal margin is negative. In the second mode, an exponential decay function reduces the maximum distance over time:

$$d_{max}(t + \Delta t) \leftarrow \begin{cases} d_{max}(t) \beta_{rmd} RSSI_{\Delta}(t) & \text{if } (d_{\Delta} < d_{thres}) \text{ or } (RSSI_{\Delta}(t) < 0) \\ \alpha_{rmd} d_{max}(t) & \text{otherwise} \end{cases} \quad (15)$$

where β_{rmd} is a user-configurable variable controlling the size of distance updates, and $\alpha_{rmd} \in [0, 1]$ controls the decay rate when the drone is outside the update region and has good signal strength. The threshold d_{thres} prevents the maximum distance from growing without bound, and the exponential decay ensures that the drone does not overestimate the communication range based on previous measurements taken under favourable propagation conditions.

3.5. Damping

The damping policy is inspired by Li et al. [24] and provides a policy acceleration in the direction opposite to the drone's travel direction, operating as a GDS in the configuration space of each drone [24]. The damping term is defined as:

$$\mathbf{B}_d(\dot{\mathbf{x}}) = \eta_d w_d \dot{\mathbf{x}}, \quad (16)$$

where η_d is a damping coefficient and w_d is a weighting coefficient. The policy acceleration is $\mathbf{f} = -\mathbf{B}_d(\dot{\mathbf{x}})$ and the priority matrix is $\mathbf{M}_d = \mathbf{G}_d = w_d \mathbf{I}$ [24]. The effect is analogous to a drag force, providing greater resistance to motion that increases with drone speed.

3.6. Policy Parameter Baseline

The policies presented in this section each depend on a set of user-configurable parameters that govern their behaviour. While these parameters can be tuned to suit specific environments and mission requirements, a baseline configuration is needed to ensure consistent and reproducible evaluation across the simulation studies in Section 5. The baseline parameters were selected through empirical tuning to achieve stable and effective drone navigation in the tested environments. The default parameters used in the motion planner are summarised in Table 1.

Table 1. Summary of baseline configuration parameters for the policies.

Policy	Parameters				
Goal attractor	α_{ga} 1	w_u 10	w_l 1	σ_γ 1	η_{ga} 2
LiDAR, obstacle avoidance	ϵ_{oa} 10^{-8}	α_{oa} 10^{-4}	η_{oa} 1		
Circle, obstacle avoidance	ϵ_{oa} 0.2	α_{oa} 10^{-5}	η_{oa} 0		
Inter-drone collision avoidance	ϵ_{oa} 0.2	α_{oa} 10^{-5}	η_{oa} 0		
Fixed maximum distance	d_{max} 2.4	w_{md} 1	α_{md} 3	β_{md} 2	γ_{md} 5
Radio, maximum distance	$RSSI_{desired}$ −50	$d_{max,init}$ 4	d_{thres} 2	β_{rmd} 0.005	α_{rmd} 0.999

4. Implementation

The simulation environments are formed as 2D spaces with three critical elements for initialisation: start position, goal position, and obstacle map. The start and goal positions are created based on the number of drones n_{drones} and a random initial start area. An obstacle map is generated based on the number of obstacles n_{obs} and a random radius between r_{obs_min} and r_{obs_max} for each circle-shaped obstacle. The simulation runs in discrete time and assumes the drone can realise the desired accelerations. Obstacles are generated without overlaps and represented as polygons with 16 vertices. Collision detection between drones and obstacles is handled using Shapely (version 1.8.1), a Python library for computational geometry.

Figure 7a shows an example of a partially specified environment with an area allocated for random obstacle generation (grey shaded area). The environment is used as a template for our simulations. The three solid red circles at the left indicate the drones' starting positions. The green solid circle on the right is the goal node, with the outer green circle being the arrival area for drones. Figure 7b shows an example of the realisation of the environment with randomly generated obstacles and drones (black dots) navigating the environment (yellow paths).

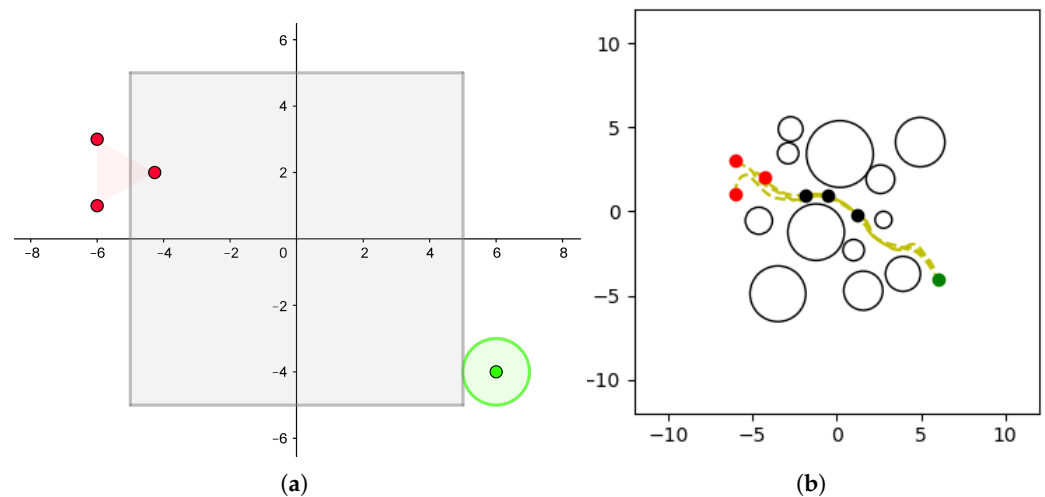


Figure 7. (a) An example of a partially defined environment template and (b) an example of a realisation with randomly generated obstacles. x and y axes are displayed in units of metres. The red solid circles indicate drones' starting positions. The green solid circles show the goal positions, with the lighter green circle indicating the goal reaching tolerance. Dashed olive green lines show the trajectories of the drones.

To sense the environment, we simulate a LiDAR sensor. The LiDAR finds the closest intersections between the line segments representing laser rays in 20 uniformly spaced directions and a polygon representation of the environment. The simulated LiDAR assumes a detection range of 2.5 m.

The reinforcement learning-based algorithm and its training environment are implemented by using Stable Baseline3 (SB3) [33]. It utilises PyTorch (version 1.11) implementations of reinforcement learning algorithms and supports the custom OpenAI Gym [34] that provides a standard API for communication between the learning agent and the environment. We applied the Proximal Policy Optimisation (PPO) algorithm, a policy gradient method introduced by [35]. Our PPO architecture was founded on an actor–critic framework. In this setup, the actor's role was to optimise and predict the most effective action, while the critic was responsible for evaluating the action based on an estimated value function. Our implementation of PPO aimed to optimise the motion policy by conservatively comparing and selecting between an old policy, $\pi_{\theta_{old}}$, and a newly updated policy π_{θ} , subject to specific constraints during each policy update. During the learning experiments, the PPO algorithm was trained with the default hyperparameters defined by SB3 PPO implementation as best practice [33]. The most critical parameters are listed in Tables 2 and 3.

Table 2. A summary of initialised PPO algorithm parameters.

PPO Parameters	
Training steps	7,000,000
Learning rate	0.0003
Maximum number of steps per episode	2048
Batch size	64
Number of epochs	10
Discount factor (Gamma)	0.99
GAE smoothing factor (Lambda)	0.95
Clip range	0.2

The parameters remain the same for all training experiments.

Hyperparameter tuning was considered beyond the scope of the current study, which may result in suboptimal trained models. Variations in the learning rate and clip range are expected to primarily affect convergence speed and training stability, while variations in reward weights are expected to affect the balance between collision avoidance and navigation efficiency. The observed improvement of the LiDAR-based residual RMP over the handcrafted baseline is expected to remain robust under different hyperparameter settings, since it is grounded in the structure of the residual learning framework rather than a specific parameter configuration. However, the absolute performance levels reported may vary with different hyperparameter settings. Systematic hyperparameter optimisation is identified as a direction for future work [36].

Table 3. Summary of RPL parameters, including observation, action, and reward/punishment gains for Case 1 (goal attractor-based) and Case 2 (LiDAR-based obstacle avoidance).

Learning Parameters	Case 1	Case 2
Observation space shape	28	22
Action space shape	5	5
Observation space range	10^{38}	10^{38}
Action space range	10^{38}	10^{38}
Reward gain for reaching goal	10	10
Punishment gain for collision	−20	−30
Reward gain for reducing the distance to the goal	1	1
Punishment gain for increasing distance to the goal	1	1
Punishment gain for high policy acceleration	0.005	0.0005
Punishment gain for accelerating without sensing obstacles	2	2
Punishment gain for the episode time steps spent	1	1
Punishment gain if the episode traversed distance is longer than the straight start to goal distance	2	2

We train in an environment with one drone in a random starting position, goal position, and five random circular obstacles. The training and simulations were performed on an Intel(R) Core(TM) i7-7700 CPU (3.60 GHz) with 32 GB DDR4 memory and integrated Intel HD Graphics 630 and a dedicated NVIDIA GeForce GTX 1080 8 GB DDR4 GPU. The operating system was Ubuntu 20.04.3 LTS. The training used the LiDAR-based residual policy, and the goal attractor residual policy was trained at approximately 95 and 400 training steps per second, respectively. The significant training efficiency difference is assumed to be caused by using the LiDAR-based obstacle avoidance policy when training the LiDAR-based residual policy.

The computational complexity of the proposed approach scales with the number of drones in the formation. The RMPflow framework itself has linear complexity $\mathcal{O}(n)$ in the number of policies in the RMP tree, due to its sparse tree structure that enables efficient message passing between nodes [14]. However, for each additional drone added to the formation, the number of pairwise inter-drone collision avoidance and maximum distance policies grows as $\mathcal{O}(n^2)$, where n is the number of drones. The overall computational complexity of the multi-drone system, therefore, scales as $\mathcal{O}(n^2)$. While this is a manageable scaling behaviour for small formations, it may impose practical limitations for larger drone swarms, and efficient policy pruning or decentralised architectures may be required to maintain real-time performance at larger scales.

5. Results and Evaluation

Table 4 summarises the selected study environments.

We simulate each obstacle avoidance policy in conjunction with a goal attractor policy on a single drone navigating 1000 randomly generated environments with five obstacles.

The random generation of obstacle sizes, positions, and drone starting positions across 1000 environments provides a substantial empirical basis for the reported averages, capturing a wide range of variability in the experimental conditions that reduces the influence of any single environment configuration on the results.

Table 4. Summary of environments used in the evaluation.

ID	Description of Environment
OA1	A single large circular obstacle between a single drone's starting position and goal.
OA2	A single rectangular obstacle between a single drone's starting position and goal position.
OA3	One drone with a random starting position, goal, and random circular obstacles. Use as a training environment.
MD2	Two drones with a rectangular obstacle that breaks the line of sight between the drones as they travel towards their goals.
OAMD3	Three drones in a random environment. Fixed starting positions and goals. Randomly generated circular obstacles. All drones have a unique goal assigned, and the formation is fully connected using maximum distance policies.

Experiments are labelled using the prefixes OA, MD, and OAMD to denote obstacle avoidance, maximum distance, and combined experiments, respectively. The numerical identifier indicates the obstacle type, where 1 denotes circular obstacles, 2 denotes rectangular obstacles and 3 denotes a set of random circles.

5.1. Learning-Enhanced Obstacle Avoidance

Figures 8–11 visualise simulations with different obstacle avoidance policies in environment OA1. In these figures, the circle (a polygon representation) refers to an obstacle in the environment. The red square and green pentagon indicate the start and goal positions, respectively. The drone is represented as “x”. The colour of the path indicates the speed. The generated collision-free paths illustrate the capability of the drones to avoid the obstacle. Figure 8 shows a path generated by the handcrafted RMPs.

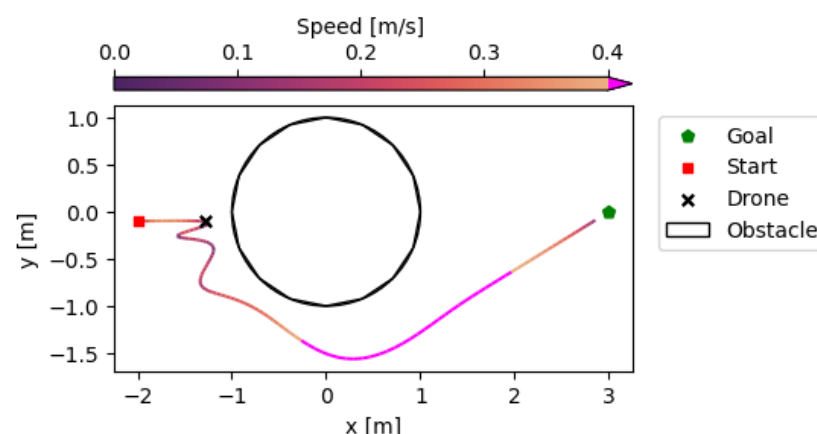


Figure 8. Simulation with circle-based obstacle avoidance policy.

The handcrafted RMP with the LiDAR model generate the path in Figure 9. It is observed that paths generated from the handcrafted RMP policy oscillate when it encounters an obstacle. Involving the LiDAR model results in a more rapid oscillation than the handcrafted RMP policy.

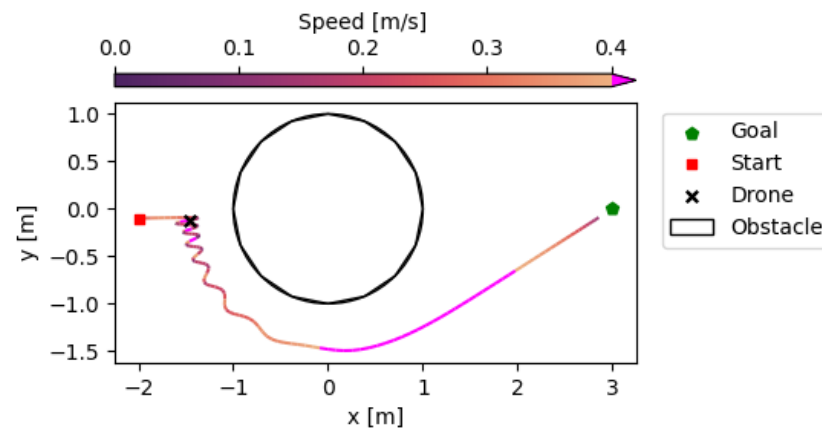


Figure 9. Traversed path with LiDAR-based obstacle avoidance policy.

Figure 10 shows a path based on residual RMP learning with the LiDAR model. Compared with Figure 9, the policy produces smaller oscillations when encountering an obstacle.

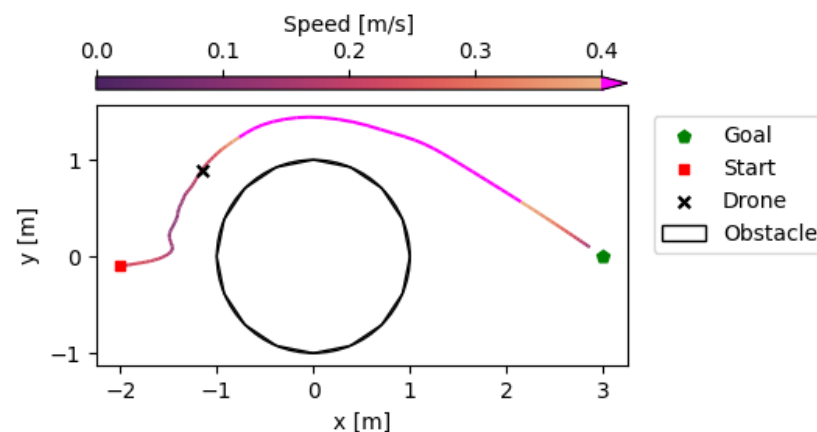


Figure 10. Traversed path with LiDAR-based residual policy.

Figure 11 shows a path generated by residual acceleration learning, which learns the residual policy of the RMP policy using universal approximation, e.g., a neural network [16]. It shows that, unlike the other obstacle avoidance policies, the residual acceleration policy does not suffer from oscillations when it encounters an obstacle. However, an abnormal path around the area of $(x, y) \simeq (1.5 \text{ m}, -0.7 \text{ m})$ was observed, likely caused by the policy converging to a local optimum during training or insufficient generalisation from the training environment to this specific obstacle configuration.

The training progression of the goal attractor-based residual policy reveals that mean episode lengths started at approximately 50 steps in the early training stages, indicating frequent collisions, before increasing to around 1400 steps after 3 million training steps. The maximum mean episode reward was reached at 1.44 million training steps, after which it decreased throughout the remaining training, and the best-performing model was found at only 390,000 training steps. This suggests that extended training did not benefit the policy and that the reward function did not provide a sufficiently reliable signal for learning obstacle avoidance from scratch. A poor correlation between the mean episode reward and the desirability of the learned behaviour was observed, consistent with findings in similar studies [16]. In contrast, the LiDAR-based residual policy only needed to refine an already functional handcrafted baseline, which substantially reduced the learning burden

and contributed to its superior performance. Full training curve plots supporting this analysis are available in [37].

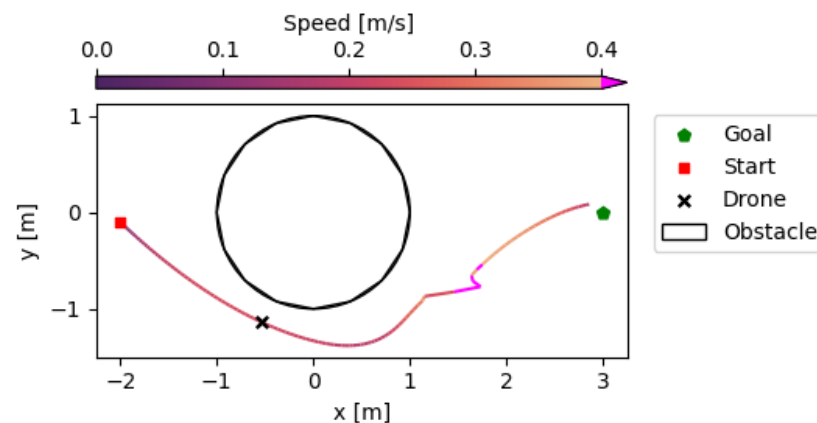


Figure 11. Traversed path with goal attractor-based residual policy generated by residual acceleration learning.

Table 5 compares the performance of the obstacle avoidance policies. It shows that the arrival time of the LiDAR-based handcrafted RMP in this environment is 25.0 s, whereas the arrival time of the LiDAR-based residual RMP is 4.4 s faster at 20.6 s. The results also show that the LiDAR-based residual RMP, on average, accelerates 0.16 m/s² less, travels 1.18 m shorter distances, and maintains approximately the same average speed as the LiDAR-based handcrafted RMP. The learned residual policies arrive earlier than the handcrafted policies, with the LiDAR-based residual RMP being the fastest.

Table 5. Summary of using each obstacle avoidance policy on environment OA1.

Policy	Success Rate [%]	Collision Rate [%]	Time of Arrival [s]	Travel Distance [m]	Travel Speed [m/s]	Acceleration [m/s ²]
Handcrafted RMP	100	0	25.0	6.79	0.26	1.08
LiDAR, handcrafted RMP	100	0	25.0	7.21	0.29	1.24
LiDAR, residual RMP	100	0	20.6	6.03	0.29	1.08
LiDAR, residual acceleration	100	0	23.8	5.91	0.26	1.28

Table 6 shows results with four different obstacle avoidance policies tested on 1000 randomly generated environments with five obstacles. The 95% confidence intervals (CI_{95%}) for the binary outcomes of success rate and collision rate are derived analytically from the binomial distribution as:

$$CI_{95\%} = \pm 1.96 \sqrt{\frac{p(1-p)}{n}}, \quad (17)$$

where p is the observed proportion of simulations in which the outcome of interest occurred, and $n = 1000$ is the number of simulations. The confidence intervals confirm that the observed differences in collision and success rates between policies are statistically significant.

Except for the success rate and collision rate, the results of the tests only reflect the performance in scenarios where the drones are successfully reaching the goal. If a drone collides, gets stuck, or, for some other reason, does not reach the destination, it counts as unsuccessfully reaching the goal. The handcrafted RMP uses circle-based obstacle avoidance. Although it shows the best performance among other obstacle avoidance policies, it holds the strong assumption that obstacles are simplified to circles, and the system knows the centre position of the obstacles. With the LiDAR-based handcrafted RMP, the drone successfully reached the goal in 90.5% of the environments. In 7.5% of

the remaining environments, it collides. Using the LiDAR-based residual RMP, the drone collides in 1.9% of environments, and the success rate is 88.6%. It outperforms the LiDAR-based handcrafted RMP on all metrics except for the success rate, which is 1.9% points lower. Especially, the collision rate is reduced by 5.6% points while on average using less time and speed to travel. The residual acceleration policy shows the worst performance, with a collision rate of 70.9%. It only succeeds in reaching the goal in 26.6% of the environments. Notably, a reduced collision rate of 74.7% is observed when a LiDAR-based residual RMP is applied compared to the handcrafted RMP.

Table 6. Results with four different obstacle avoidance policies were tested on 1000 random environments with five obstacles in environment OA3. Average numbers from 1000 simulations are listed.

Policy	Success Rate [%]	Collision Rate [%]	Time of Arrival [s]	Travel Distance [m]	Speed [m/s]	Acceleration [m/s ²]
Handcrafted RMP	90.0 ± 1.9	0 ± 0.0	19.9	7.92	1.40	1.06
LiDAR, handcrafted RMP	90.5 ± 1.8	7.5 ± 1.6	22.6	8.81	1.40	1.10
LiDAR, residual RMP	88.6 ± 2.0	1.9 ± 0.8	21.3	8.05	1.40	1.08
LiDAR, residual acceleration	26.6 ± 2.7	70.9 ± 2.8	41.8	22.9	1.65	1.96

The handcrafted RMP shows significant results with no collisions and the shortest travel distance of 7.92 m on average when the LiDAR model is integrated into the LiDAR-based handcrafted RMP. This results in a collision rate and a delayed average time of arrival of only 19 s. However, this policy is rather naive since it is a prerequisite to know the sizes and positions of the circle-shaped obstacles for using the policy. This indicates the demand for further improving the handcrafted policy parameters. To improve motion policies automatically, we involve the residual policy mechanism. After training, the LiDAR-based residual RMP reduced the arrival time and travelled distance compared with the LiDAR-based handcrafted RMP. The success rate is 2.6% points lower than the handcrafted LiDAR-based policy, likely attributable to a directional bias introduced during training, where the policy learned a preference for turning in a particular direction around obstacles. While this bias reduces collisions in most scenarios, it may cause the drone to take suboptimal paths in specific obstacle configurations, occasionally leading to the drone getting stuck or failing to reach the goal. This suggests that the training environment OA3 may not have exposed the policy to a sufficiently diverse range of obstacle approach angles to eliminate directional bias. Using the residual acceleration shows high uncertainty to generalise the method for different obstacle environments, i.e., 70.9% collision rate. Learning obstacle avoidance from scratch (universal approximation) with the goal attractor policy, therefore, requires more training effort than using a residual policy mechanism. However, it shows more flexibility in shaping the behaviour, e.g., less oscillation in some specific environments.

5.2. Communication Maintenance

The simulation environment MD2 is designed with an obstacle obstructing the line of sight for a pair of drones. It evaluates the radio-based maximum distance policy with changing radio propagation conditions. In this scenario, the motion policies for drones comprise a goal attractor, LiDAR-based handcrafted RMP obstacle avoidance, and radio-based maximum distance policies.

Figure 12 shows when two drones start losing line of sight between each other at the simulation time $t = 12$ s. The maximum allowed distances are illustrated with circles.

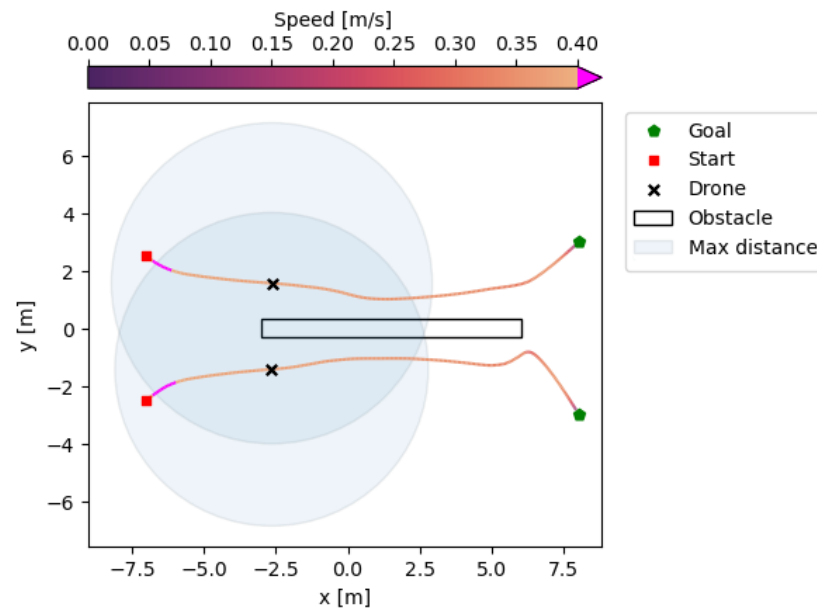


Figure 12. Simulation of drones in the environment MD2 with radio-based maximum distance policy active at the point where the obstacle obstructs the line of sight. The light-blue solid circles illustrate the range of communication.

The figure also shows when two drones are closing to regain the line of sight at the simulation time $t = 35$ s. Figure 13 shows that the drones are within each other's desired maximum distances, while their paths bend towards the obstacle once it obstructs the line of sight.

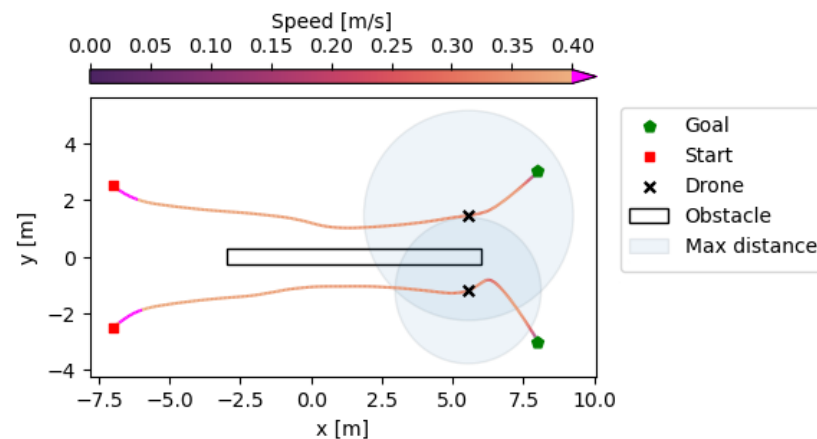


Figure 13. Simulation of environment MD2 with the maximum allowed distances illustrated for the two drones after travelling along an obstacle obstructing the line of sight in environment MD2.

Figure 14 shows that the maximum allowed distances decrease when the line of sight is obstructed and increase once the line of sight is restored. The green curve shows the actual distance between the drones.

Figure 15 indicates that the RSSI drops when the line of sight is obstructed and is stabilised to a desired minimum RSSI level by shortening the distance between drones. Vertical lines in Figure 15 mark these two specific moments, which present the maximum allowed distances and RSSI values for two drones travelling along the paths.

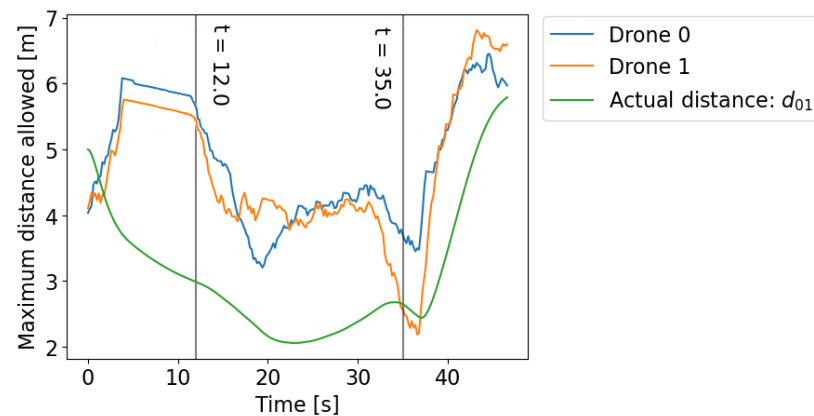


Figure 14. Maximum allowed distance for two drones travelling along an object obstructing the line of sight in environment MD2.

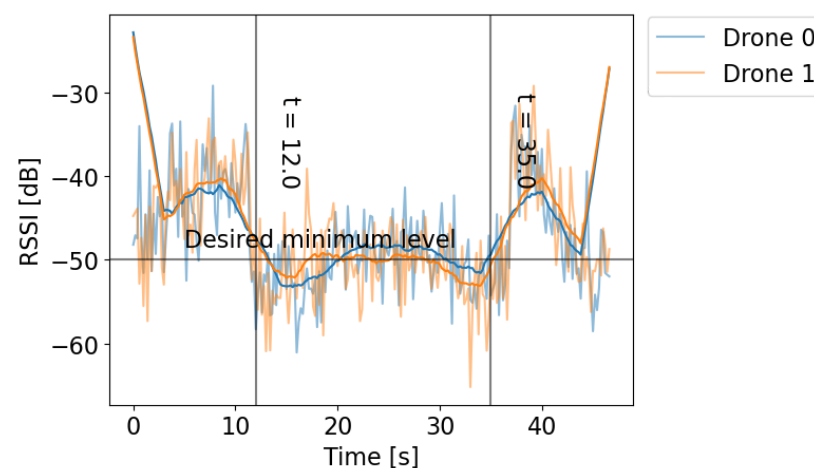


Figure 15. RSSI values for two drones travelling along an object obstructing the line of sight in environment MD2.

5.3. Obstacle Avoidance with Communication Maintenance

An evaluation is constructed for various motion policies that guide three drones toward goal positions while maintaining a maximum distance between drones and avoiding obstacles. The drones are initialised in every simulation with the same start and goal positions. Five randomly-sized circle-shaped obstacles are generated and placed in the simulation environment OAMD3 (Figure 16).

The fixed maximum distance and radio-based maximum distance policies are tested in combination with LiDAR-based handcrafted RMP or LiDAR-based residual RMP for obstacle avoidance. The results are summarised in Table 7. The 95% confidence intervals ($CI_{95\%}$) are computed using the binomial formula (Equation (17)). In addition, the goal attractor policy and inter-drone collision avoidance policy are included in all tests. Each of the four results in Table 7 is based on simulations in 1000 environments. Compared with the LiDAR-based handcrafted RMP policy, the LiDAR-based residual RMP policy reduces the collision rate by 9.1% points using the fixed maximum distance policy and 8.3% points using the radio-based maximum distance policy. This corresponds to 35.5% and 37.9%, respectively.

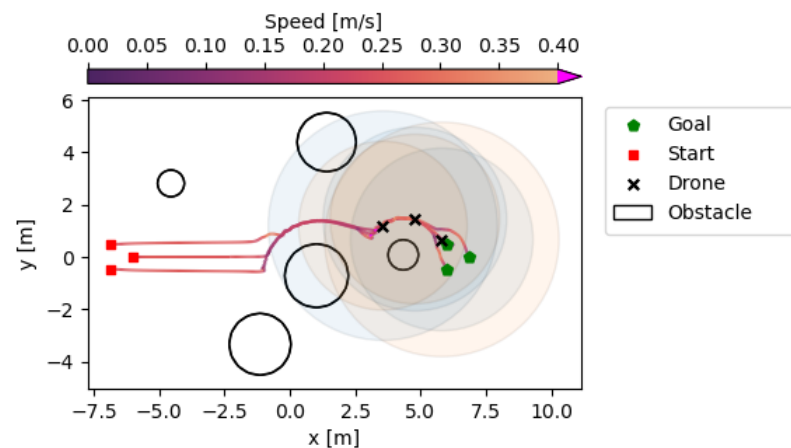


Figure 16. Simulation realising environment OAMD3 with LiDAR-based obstacle avoidance and radio-based maximum distance policies. The maximum communication distance for each drone is illustrated with a coloured circle.

Table 7. Results from running environment OAMD3. Average numbers of 1000 simulations are listed.

Policy	Arrival Success Rate [%]	Collision Rate [%]	Time of Arrival [s]	Travel Distance [m]	Speed [m/s]	Acceleration [m/s ²]
LiDAR-based handcrafted RMP for obstacle avoidance						
Fixed maximum distance	70.3 ± 2.8	25.7 ± 2.7	113.53	14.33	1.13	1.03
Radio, maximum distance	76.7 ± 2.6	21.9 ± 2.6	90.43	13.95	1.16	1.03
LiDAR-based residual RMP for obstacle avoidance						
Fixed maximum distance	77.0 ± 2.6	16.6 ± 2.3	113.38	14.71	1.14	1.03
Radio, maximum distance	82.0 ± 2.4	13.6 ± 2.1	85.32	14.19	1.17	1.03

The radio-based maximum distance policy results in fewer collisions and reduces the time of arrival compared to the fixed maximum distance policy. Although the LiDAR-based residual RMP is not trained with maximum distance policies, the improvement persists when the learned policy is combined with maximum distance policies.

The radio-based maximum distance policy outperforms the fixed maximum distance policy. This is evidenced by the reduction of the travel time by 20.3% and 24.7% in the tests. This improvement reflects applying a conservative worst-case scenario for the maximum distance parameter in the fixed maximum distance policy. As a result, the conservative fixed distance typically results in a smaller allowed range compared to the radio-based maximum distance policy, which generally permits greater distances. In addition, the radio-based maximum distance policy shows fluctuations in the maximum allowed distance. The effect of the fluctuations could be isolated and examined by setting the fixed maximum distance to the average distance of the radio-based maximum distance policy.

6. Discussion

The present work constitutes a necessary foundational step toward a fully deployable multi-drone motion planning framework. The results are therefore to be interpreted in that context, and the limitations discussed below bound the conclusions that can be drawn from the current work.

The proposed approach is designed to address a specific set of challenges in multi-drone navigation, namely the combination of reactive obstacle avoidance, formation cohesion, and communication-aware distance control in unknown environments. To contextualise the properties of the proposed approach relative to mainstream motion planning

methods, Table 8 provides a qualitative comparison across key properties relevant to multi-drone navigation in unknown environments.

Table 8. Qualitative comparison of motion planning methods.

Method	Environment Knowledge	Computational Load	Multi-Objective Composition	Communication-Aware	Unknown Environments
MPC [7,28]	Yes	High	Limited	No	No
APF [8]	Partial	Low	Limited	No	Yes
ORCA [10]	Partial	Low	No	No	Yes
DWA [9]	Partial	Low	No	No	Yes
RL [13,26]	No	High (training)	Yes	No	Yes
RMP (proposed)	No	Low	Yes	Yes	Yes

The table illustrates that the proposed approach is the only method among those considered that combines low computational load, multi-objective composition, communication-awareness, and applicability to unknown environments without requiring prior environment knowledge.

The LiDAR-based residual policy improved compared to the handcrafted LiDAR-based obstacle policy. This indicates that reinforcement learning can improve sub-optimal policies. The improvement is mainly a reduction in the average number of collisions from a 35.3% reduction for drones flying in formation to a 74.7% reduction for drones flying alone in an environment of random obstacles. The intention was to improve the time of arrival and the travelled distance. Rigorous and structured parameter tuning could have produced a better configuration of the LiDAR-based obstacle avoidance before training a neural network to improve the policy, thereby mitigating collisions. Training the goal attractor-based residual policy, which should learn obstacle avoidance from scratch, did not satisfactorily learn the desired motion behaviour (collides in 70% of the simulated scenarios). The undesired performance of using the goal attractor residual policy is unexpected. The policy is trained in a similar approach as [18,19], which shows promising performance by achieving a substantially lower collision rate. However, our learning-based policy results indicate that learning a few capabilities is more likely to perform well than learning multiple. Similar trends are seen in [18]. The performance of the learned models can likely improve by using a hyperparameter tuning of both PPO and the reward function parameters. However, the reward function does not give a good indication of whether the behaviour is desirable or not, and no alternative metric has been found that describes the behaviour unambiguously. Therefore, a systematic automated hyperparameter optimisation was not performed.

The simulation of the maximum distance policies assumes that the positions of all connected drones are exchanged at each step of the simulation. This naturally puts a high load on the communication network. Reducing the rate of messages transmitted between the drones has not been examined. Moreover, the simulation does not account for a physical model for the drones but exclusively concentrates on the behaviour of the RMPs. The lack of a physical model allows the system to make manoeuvres that may be infeasible in a real setting. In particular, real-world constraints such as actuator saturation, attitude dynamics, control delays, and vehicle inertia may limit the ability of the drone to realise the commanded accelerations, potentially degrading obstacle avoidance and formation performance. These limitations could be addressed in future work by incorporating a quadrotor dynamics model to constrain accelerations to physically realisable values, and by adopting an event-triggered communication scheme to reduce the load on the communication network through selective message transmission [28].

The simulation works at a relatively low sampling rate of 5 Hz, which is insufficient for real drone deployment, where significantly faster update rates are typically required for reliable reactive obstacle avoidance. However, the low sampling rate was chosen out of concern for the computational load while training the learned policies in the context of the 2D simulation study. The low sampling rate may have adverse effects on the system's stability. If the desired accelerations of the RMPs have steep falloffs, the drone may travel a considerable distance between time steps where the desired accelerations change considerably. Achieving higher update rates, alongside the extension to 3D simulations with quadrotor dynamics, is identified as a direction for future work.

The radio path loss simulation is somewhat simplified. It uses a free space path loss channel model with random variation to model small-scale path losses and attenuation in non-line-of-sight situations. The large-scale path loss could, without major impacts on the simulation speed, be based on more accurate empirical models applying more realistic path loss exponents characteristic of the different environments [32]. Because the experiments with the radio-based maximum distance policy are abstract, it is difficult to justify using any particular empirical model over another. Thus, the experiments with the radio-based maximum distance policy are not directly transferable to a real-life setting but indicate only a typical behaviour.

The current communication model does not account for multipath fading, packet loss, latency, interference, or bandwidth limitations. These factors could adversely affect the effectiveness of the radio-based maximum distance policy in realistic environments. Specifically, multipath fading and interference may cause rapid fluctuations in the measured RSSI, leading to unstable updates of the maximum allowed distance and potentially compromising formation stability. Packet loss and latency may delay the exchange of drone positions, reducing the responsiveness of the formation control policy. The results should therefore be interpreted as a proof-of-concept demonstration of the radio-aware formation control principle rather than a fully realistic communication simulation. More realistic communication modelling, including multipath fading and channel dynamics in drone propagation scenarios, is identified as a direction for future work, building on recent work by the authors on ground-to-air channel characterisation [38].

Occasionally, the drones get stuck in the environment due to concave-like constellations of obstacles. Failing to navigate around concave structures does not indicate a failure on the part of the policies, as these are only intended for short-term motion planning. Navigation around concave obstacles could be improved by global motion planning.

Future Research Directions

The present work is deliberately scoped to two-dimensional environments for two reasons. First, the target application of infrastructure inspection of linear infrastructure, such as railway catenary lines and bridges [1], involves largely planar flight paths, making the 2D assumption well-motivated for this context. Second, the 2D setting substantially reduces the computational cost of reinforcement learning training, enabling a more systematic evaluation of policy compositions. Future work, therefore, includes 3D simulations with drone dynamics and verifying policies on real drones. Real-world validation of RMP-based planners on drone platforms has been demonstrated within the research group [22], providing a foundation for future hardware validation of the proposed framework. Deploying the proposed framework on a practical drone platform would require integration with a flight controller such as PX4 or ArduPilot, with the RMPflow planner operating as an offboard controller providing velocity or acceleration setpoints. Key implementation challenges include achieving sufficiently high update rates for reactive obstacle avoidance, ensuring reliable LiDAR data acquisition and processing within the control loop, and validating the

radio-based maximum distance policy under realistic wireless channel conditions. The feasibility of deploying RMP-based reactive planners on real drone platforms has been demonstrated in [22], where a similar reactive planner based on Forced Geometric Fabrics was validated in real-world experiments at 20 Hz using only approximately 5–7% of a single CPU core. More realistic radio models and link parameters could refine the maximum distance policy. Larger drone swarm formations should also be studied, considering data transmission needs and relevant network protocols.

In addition, we plan to enhance our simulation model to include perturbing effects of the environment, such as wind, that affect the drones' motion, as well as implement a more refined obstacle model that includes both static and dynamic obstacles with mixed shapes. The study presented in [39] highlights the importance of accurately modelling disruptions in multi-drone simulations. By addressing environmental and operational challenges, the work advances drone swarm resiliency, ensuring safer and more reliable deployment. Furthermore, a formal stability analysis of the combined residual and handcrafted RMP policies, including conditions under which the combined policy preserves geometric consistency within the RMPflow composition, is identified as an important direction for future work.

Future work will focus on enhancing the reinforcement learning framework by integrating global motion planning algorithms such as A* or RRT with learned policies. This hybrid approach aims to address the limitations of the current reward function, which may not fully capture the desired behaviours and can slow down the learning process. By incorporating global planning, the system can navigate complex scenarios, such as concave obstructions, more effectively, improving both learning efficiency and the overall performance of drone formations in dynamic environments. Additionally, a formal ablation study systematically evaluating the contribution of individual reward components to the learned behaviour would help identify redundant or counterproductive terms and further guide reward function design.

Furthermore, the performance of the residual learning policies could be improved through systematic optimisation of the PPO hyperparameters, such as the learning rate, clipping parameter, and entropy coefficient. PPO is known to be sensitive to hyperparameter settings, and suboptimal configurations can lead to premature convergence to suboptimal policies [36]. Automated hyperparameter search methods such as Bayesian optimisation or population-based training could be employed to identify configurations that improve both learning efficiency and final policy performance, particularly in multi-drone formation scenarios where the training environment is more complex. Additionally, a more rigorous quantitative characterisation of oscillatory behaviour, using metrics such as jerk, trajectory smoothness, or heading variance, would provide a more objective basis for evaluating the improvement achieved by the residual learning policies and is identified as a direction for future work.

7. Conclusions

In this paper, we investigated using a set of Riemannian Motion Policies (RMPs) constructed for the RMPflow framework for effective multi-drone motion planning. The proposed RMPs serve the purpose of avoiding obstacles and maintaining a flexible formation. Three RMPs were designed for obstacle avoidance in unknown environments. The first RMP is a handcrafted LiDAR-based obstacle avoidance policy. The second RMP is a LiDAR-based residual policy that augments the LiDAR-based obstacle avoidance policy with a neural network trained using reinforcement learning. The third RMP is a goal attractor-based residual policy that augments a goal attractor policy trained with reinforcement learning. Two different RMPs were created to maintain a flexible drone

formation. A fixed maximum distance policy is designed and configured to keep drones within radio communication range. The other, radio-based maximum distance policy, is a dynamically changing maximum distance to adapt to varying radio propagation conditions. The policies were evaluated by simulations in different 2D environments with obstacles. When simulating, the proposed RMPs are combined with existing RMPs in the RMPflow framework for reaching goals, inter-drone collision avoidance, and acceleration damping. We have demonstrated that reinforcement learning can improve a sub-optimal manually defined obstacle avoidance policy with collision rates of formations and single drones reduced by 35.4% and 74.7% less, respectively.

Formation flying with three drones using the maximum distance policies was tested in 1000 randomly generated environments with five obstacles. Compared to the fixed maximum distance policy, the radio-based maximum distance policy provided a more flexible maximum distance between drones, effectively improving the total drone swarm performance when navigating through an unknown environment. Notably, it improves the time a drone formation spends on navigating a cluttered environment by 20% to 25% in simulations when combined with the LiDAR-based obstacle avoidance and LiDAR-based residual policies, respectively.

Author Contributions: Conceptualisation, L.S., E.H.S. and A.L.C.; Funding acquisition, R.H.J.; Methodology, E.H.S. and A.L.C.; Project administration, R.H.J.; Resources, R.H.J.; Software, E.H.S. and A.L.C.; Supervision, L.S. and R.H.J.; Validation, L.S., E.H.S. and A.L.C.; Visualization, E.H.S. and A.L.C.; Writing—original draft, E.H.S. and A.L.C.; Writing—review & editing, L.S. and R.H.J. All authors have read and agreed to the published version of the manuscript.

Funding: This project has received funding from the European Union’s Horizon 2020 research and innovation project Drones4Safety under grant agreement No 861111.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No specific data set resulted from the study. The source code for the simulations is available on GitHub (version 1.0): https://github.com/esbenhaubroskov/master_project/tree/main/RMP/rmp-project (accessed on 3 April 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

2D	Two-dimensional
3D	Three-dimensional
A*	A star
APF	Artificial Potential Field
API	Application Programming Interface
DWA	Dynamic Window Approach
GAE	Generalised Advantage Estimation
GDS	Geometric Dynamical System
GPU	Graphics Processing Unit
LiDAR	Light Detection And Ranging
MAV	Micro Aerial Vehicle
MD	Maximum Distance
MPC	Model Predictive Control
OA	Obstacle Avoidance
ORCA	Optimal Reciprocal Collision Avoidance
PPO	Proximal Policy Optimisation

RL	Reinforcement Learning
RPL	Residual Policy Learning
RMP	Riemannian Motion Policy
RSSI	Received Signal Strength Indicator
RRT	Rapidly-exploring Random Tree
SB3	Stable Baseline3
UAV	Unmanned Aerial Vehicle

References

- Jacobsen, R.H.; Matlekovic, L.; Shi, L.; Malle, N.; Ayoub, N.; Hageman, K.; Hansen, S.; Nyboe, F.F.; Ebeid, E. Design of an Autonomous Cooperative Drone Swarm for Inspections of Safety Critical Infrastructure. *Appl. Sci.* **2023**, *13*, 1256. <https://doi.org/10.3390/app13031256>.
- Asaamoning, G.; Mendes, P.; Rosário, D.; Cerqueira, E. Drone Swarms as Networked Control Systems by Integration of Networking and Computing. *Sensors* **2021**, *21*, 2642. <https://doi.org/10.3390/s21082642>.
- Elmeseiry, N.; Alshaer, N.; Ismail, T. A Detailed Survey and Future Directions of Unmanned Aerial Vehicles (UAVs) with Potential Applications. *Aerospace* **2021**, *8*, 363. <https://doi.org/10.3390/aerospace8120363>.
- Shi, L.; Mehrooz, G.; Jacobsen, R.H. Inspection Path Planning for Aerial Vehicles via Sampling-based Sequential Optimization. In Proceedings of the 2021 International Conference on Unmanned Aircraft Systems (ICUAS), Athens, Greece, 15–18 June 2021; pp. 679–687. <https://doi.org/10.1109/ICUAS51884.2021.9476784>.
- Zhou, C.; Huang, B.; Fränti, P. A review of motion planning algorithms for intelligent robots. *J. Intell. Manuf.* **2022**, *33*, 387–424.
- LaValle, S.M.; Kuffner, J.J., Jr. Randomized kinodynamic planning. *Int. J. Robot. Res.* **2001**, *20*, 378–400.
- Richards, A.; How, J. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301), Online, 8–10 May 2002; Volume 3, pp. 1936–1941. <https://doi.org/10.1109/ACC.2002.1023918>.
- Khatib, O. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *Int. J. Robot. Res.* **1986**, *5*, 90–98. <https://doi.org/10.1177/027836498600500106>.
- Fox, D.; Burgard, W.; Thrun, S. The dynamic window approach to collision avoidance. *IEEE Robot. Autom. Mag.* **1997**, *4*, 23–33. <https://doi.org/10.1109/100.580977>.
- van den Berg, J.; Guy, S.J.; Lin, M.; Manocha, D. Reciprocal n-Body Collision Avoidance. In *Proceedings of the Robotics Research*; Pradalier, C., Siegwart, R., Hirzinger, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 3–19.
- Ratliff, N.D.; Issac, J.; Kappler, D.; Birchfield, S.; Fox, D. Riemannian Motion Policies. *arXiv* **2018**, arXiv:1801.02854.
- Cheng, C.A.; Mukadam, M.; Issac, J.; Birchfield, S.; Fox, D.; Boots, B.; Ratliff, N. RMPflow: A Geometric Framework for Generation of Multitask Motion Policies. *IEEE Trans. Autom. Sci. Eng.* **2021**, *18*, 968–987. <https://doi.org/10.1109/TASE.2021.3053422>.
- La, H.M.; Lim, R.; Sheng, W. Multirobot Cooperative Learning for Predator Avoidance. *IEEE Trans. Control Syst. Technol.* **2015**, *23*, 52–63. <https://doi.org/10.1109/TCST.2014.2312392>.
- Cheng, C.A.; Mukadam, M.; Issac, J.; Birchfield, S.; Fox, D.; Boots, B.; Ratliff, N. RMPflow: A Computational Graph for Automatic Motion Policy Generation. *Springer Proc. Adv. Robot.* **2020**, *14*, 441–457. https://doi.org/10.1007/978-3-030-44051-0_26.
- Xie, M.; Wyk, K.V.; Li, A.; Rana, M.A.; Wan, Q.; Fox, D.; Boots, B.; Ratliff, N. Geometric Fabrics for the Acceleration-based Design of Robotic Motion. *arXiv* **2021**, arXiv:2010.14750.
- Johannink, T.; Bahl, S.; Nair, A.; Luo, J.; Kumar, A.; Loskyll, M.; Ojea, J.A.; Solowjow, E.; Levine, S. Residual Reinforcement Learning for Robot Control. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 6023–6029. <https://doi.org/10.1109/ICRA.2019.8794127>.
- Silver, T.; Allen, K.; Tenenbaum, J.; Kaelbling, L. Residual Policy Learning. *arXiv* **2019**, arXiv:1812.06298.
- Aljalbout, E.; Chen, J.; Ritt, K.; Ulmer, M.; Haddadin, S. Learning Vision-based Reactive Policies for Obstacle Avoidance. In *Proceedings of the 2020 Conference on Robot Learning*; Kober, J., Ramos, F., Tomlin, C., Eds.; Proceedings of Machine Learning Research (PMLR): New York, NY, USA, 2021; Volume 155, pp. 2040–2054.
- Li, A.; Cheng, C.A.; Rana, M.A.; Xie, M.; Van Wyk, K.; Ratliff, N.; Boots, B. RMP2: A Structured Composable Policy Class for Robot Learning. *arXiv* **2021**, arXiv:2103.05922.
- Meng, X.; Ratliff, N.; Xiang, Y.; Fox, D. Neural Autonomous Navigation with Riemannian Motion Policy. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 8860–8866. <https://doi.org/10.1109/ICRA.2019.8794223>.
- Pantic, M.; Meijer, I.; Bähmann, R.; Alatur, N.; Andersson, O.; Cadena, C.; Siegwart, R.; Ott, L. Obstacle Avoidance Using Raycasting and Riemannian Motion Policies at kHz Rates for MAVs. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023.

22. Shi, L.; Pantic, M.; Andersson, O.; Tognon, M.; Siegwart, R.; Jacobsen, R.H. Reactive Motion Planning for Rope Manipulation and Collision Avoidance using Aerial Robots. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 3384–3391. <https://doi.org/10.1109/IROS47612.2022.9981658>.
23. Liu, Y.; Bucknall, R. A survey of formation control and motion planning of multiple unmanned vehicles. *Robotica* **2018**, *36*, 1019–1047. <https://doi.org/10.1017/S0263574718000218>.
24. Li, A.; Mukadam, M.; Egerstedt, M.; Boots, B. Multi-objective Policy Generation for Multi-robot Systems Using Riemannian Motion Policies. In *Proceedings of the Robotics Research*; Asfour, T., Yoshida, E., Park, J., Christensen, H., Khatib, O., Eds.; Springer: Cham, Switzerland, 2022; pp. 258–274.
25. Li, A.; Luo, W.; Nagavalli, S.; Sycara, K. Decentralized coordinated motion for a large team of robots preserving connectivity and avoiding collisions. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 1505–1511. <https://doi.org/10.1109/ICRA.2017.7989180>.
26. Xi, M.; Dai, H.; He, J.; Li, W.; Wen, J.; Xiao, S.; Yang, J. A Lightweight Reinforcement Learning-Based Real-Time Path Planning Method for Unmanned Aerial Vehicles. *IEEE Internet Things J.* **2024**, *11*, 21061–21071. <https://doi.org/10.1109/JIOT.2024.3350525>.
27. Shi, L.; Hernández Marcano, N.J.; Jacobsen, R.H. Multi-UAV Path Coordination through Generalized Potential Games. In Proceedings of the 2022 8th International Conference on Automation, Robotics and Applications (ICARA), Virtually, 18–20 February 2022; pp. 125–129. <https://doi.org/10.1109/ICARA55094.2022.9738578>.
28. Gräfe, A.; Eickhoff, J.; Trimpe, S. Event-triggered and distributed model predictive control for guaranteed collision avoidance in UAV swarms. *IFAC-PapersOnLine* **2022**, *55*, 79–84. <https://doi.org/10.1016/j.ifacol.2022.07.239>.
29. Ratliff, N.D.; Wyk, K.V.; Xie, M.; Li, A.; Rana, M.A. Optimization Fabrics for Behavioral Design. *arXiv* **2021**, arXiv:2010.15676.
30. Rana, M.A.; Li, A.; Ravichandar, H.; Mukadam, M.; Chernova, S.; Fox, D.; Boots, B.; Ratliff, N. Learning Reactive Motion Policies in Multiple Task Spaces from Human Demonstrations. In *Proceedings of the Conference on Robot Learning*; PMLR: New York, NY, USA, 2020; Volume 100, pp. 1457–1468.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778. <https://doi.org/10.1109/CVPR.2016.90>.
32. Hernández Marcano, N.J.; Brask, C.G.; Shi, L.; Jacobsen, R.H. An Experimental Study of 2.4 GHz LoRa in Air-to-Ground Communication. In Proceedings of the 2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall), Washington, DC, USA, 7–10 October 2024; pp. 1–6.
33. Stable Baselines3 Contributors. Source Code for Stable_Baselines3.ppo.ppo. Available online: https://stable-baselines3.readthedocs.io/en/master/_modules/stable_baselines3/ppo/ppo.html (accessed on 19 May 2022).
34. Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; Zaremba, W. OpenAI Gym. *arXiv* **2016**, arXiv:1606.01540.
35. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
36. Zhang, J.; Zhang, Z.; Han, S.; Lü, S. Proximal policy optimization via enhanced exploration efficiency. *Inf. Sci.* **2022**, *609*, 750–765. <https://doi.org/10.1016/j.ins.2022.07.111>.
37. Skov, E.H.; Christensen, A.L. Learning Enhanced Reactive Motion Planning for Obstacle Avoidance and Communication Maintenance. Master’s Thesis, Aarhus University, Aarhus, Denmark, 2023.
38. Khawaja, W.; Jacobsen, R.H.; Hussain, S.; Guvenc, I. Predicting Lifespan of Ground-to-Air Multipath Components in mmWave UAV Channels. In Proceedings of the 2025 IEEE Wireless Communications and Networking Conference (WCNC), Milan, Italy, 24–27 March 2025.
39. Phadke, A.; Medrano, F.A.; Chu, T.; Sekharan, C.N.; Starek, M.J. Modeling Wind and Obstacle Disturbances for Effective Performance Observations and Analysis of Resilience in UAV Swarms. *Aerospace* **2024**, *11*, 237. <https://doi.org/10.3390/aerospace11030237>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.