Czech Technical University in Prague

Faculty of Electrical Engineering
Department of Cybernetics
Multi-robot Systems

Development of a Safe Flocking Algorithm for UAVs Using 3D Lidar and Collaborative Multi-Robot Coordination

Bachelor's Thesis

Michal Kamler

Prague, May 2025

Study programme: Electrical Engineering and Information Technology
Branch of study: Cybernetics and Robotics

Supervisor: Ing. Manuel Boldrer, Ph.D.

Acknowledgments

Firstly, I would like to express my gratitude to my supervisor.

ČVUT FEL

Katedra kybernetiky

Datum vytvoření návrhu ZP: 22.01.2025

Typ práce: Bakalářská

Návrh zadání závěrečné práce

Vyplněný formulář včetně podpisu vedoucího práce předejte studijní referentce katedry kybernetiky, Janě Zichové (KN:E-212). Elektronickou verzi zašlete na zichova@fel.cvut.cz.

Jméno a příjmení studenta: Michal Kamler

Program: Kybernetika a robotika **Specializace:**

Email: kamlemic@fel.cvut.cz **Telefon:** 722 001 775

Název tématu česky: Vývoj bezpečného algoritmu pro koordinaci UAV pomocí 3D lidaru a koordinace více robotů

Název tématu anglicky: Development of a Safe Flocking Algorithm for UAVs Using 3D Lidar and Collaborative Multi-Robot Coordination

Práce bude vypracována v jazyce: anglickém

Zadání dle požadavků průmyslu, nebo vedoucí, konzultant, či oponent je z průmyslu: NE

Pokyny pro vypracování:

Uveďte ve výše specifikovaném jazyce. Řiďte se Požadavky na zadání ZP. Pokyny by měly obsahovat: (1) jasně definovaný softwarový nebo výzkumný cíl; (2) požadavek na analýzu existujících relevantních metod, algoritmů, přístupů nebo technologií; (3) konkrétní požadavek na kreativní komponentu ZP (navrhni, formalizuj, naprogramuj, sestroj); (4) požadavek na zhodnocení výsledku práce buď (i) pomocí teoretické analýzy a formálních důkazů a/nebo (ii) pomocí měřitelného empirického zhodnocení na relevantních datových sadách nebo testovacích scénářích (jejichž specifikace by měla být součást zadání) a porovnání vůči stavu před vyřešením problému. Požadované výstupy ZP musí být definovány tak, aby státnicová komise mohla jednoznačně vyhodnotit splnění zadání i bez přítomnosti vedoucího a oponenta u obhajoby.

(1) The research will be focused on developing a safe flocking algorithm for UAVs in C++ within the MRS system framework . The algorithm must ensure collision-free and efficient movement of UAV. (2) Some of the most promising algorithms in the literature will be compared within the MRS
simulator, in particular [1],[2],[3]. (3) The novel contributions of the thesis will include

i) Extension of existing multi-robot algorithm from 2d to 3d.
ii) Usage of 3d lidar to localize, sense process and react to environments such as a forest.
iii) Researching on possible enhancements of the algorithm for example using neural networks or learning-based techniques.

(4) An Experimental campaign to validate the theoretical findings will be conducted. This will involve simulating different real-world scenarios, such as navigating through forests or crowded spaces.

Vedoucí práce: Manuel Boldrer

Garant za katedru: **Podpis garanta:**

Vyplňuje se pouze v případě, že vedoucí práce je doktorand nebo externista (tj. není zaměstnancem FEL), viz osoba garanta.

Navržený oponent: Jan Faigl

Oponent diplomové práce nemá být členem stejné katedry jako vedoucí. Oponent bakalářské práce nemá být členem stejného oddělení katedry jako vedoucí. Je-li vedoucím doktorand ve studijní etapě, oponentem musí být někdo zkušenější než je vedoucí. Viz osoba oponenta.

Podpis vedoucího práce: **Podpis garanta specializace:** **Podpis vedoucího katedry:**

Declaration

I declare that presented work was developed independently, and that I have listed all sources of information used within, in accordance with the Methodical instructions for observing ethical principles in preparation of university theses.

Date ....

Abstract

The study of autonomous Unmanned Aerial Vehicles (UAVs) has become a prominent sub-field of mobile robotics.

Keywords TODOUnmanned Aerial Vehicles, Automatic Control

Abbreviations

UAV Unmanned Aerial Vehicle

Contents

1 Introduction

TODO

1.1 Related works

TODO

1.2 Contributions

TODO

1.3 Mathematical notation

TODO

$\mathcal{N}_i$	set of neighboring agents for the i-th agent
$\mathbf{x}$, $\boldsymbol{\alpha}$	vector, pseudo-vector, or tuple
$\hat{\mathbf{x}}$, $\hat{\boldsymbol{\omega}}$	unit vector or unit pseudo-vector
$\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3$	elements of the *standard basis*
$\mathbf{X}, \boldsymbol{\Omega}$	matrix
$\mathbf{I}$	identity matrix
$x = \mathbf{a}^\intercal\mathbf{b}$	inner product of $\mathbf{a}$, $\mathbf{b} \in \mathbb{R}^3$
$\mathbf{x} = \mathbf{a} \times \mathbf{b}$	cross product of $\mathbf{a}$, $\mathbf{b} \in \mathbb{R}^3$
$\mathbf{x} = \mathbf{a} \circ \mathbf{b}$	element-wise product of $\mathbf{a}$, $\mathbf{b} \in \mathbb{R}^3$
$\mathbf{x}_{(n)} = \mathbf{x}^\intercal\hat{\mathbf{e}}_n$	n^{th} vector element (row), $\mathbf{x}, \mathbf{e} \in \mathbb{R}^3$
$\mathbf{X}_{(a,b)}$	matrix element, (row, column)
x_d	x_d is *desired*, a reference
$\dot{x}, \ddot{x}, \dddot{x}, \ddddot{x}$	1^{st}, 2^{nd}, 3^{rd}, and 4^{th} time derivative of x
$x_{[n]}$	x at the sample n
$\mathbf{A}, \mathbf{B}, \mathbf{x}$	LTI system matrix, input matrix and input vector
SO(3)	3D special orthogonal group of rotations
SE(3)	*SO(3)* $\times \mathbb{R}^3$, special Euclidean group

Table 1.1: Mathematical notation, nomenclature and notable symbols.

2 Extension of the Rule-Based Lloyd Algorithm to 3D

2.1 Introduction to the RBL

This chapter will contain overview of the RBL algo in 2d, with some key principles for motion planning, safety and convergece. Also some applications and limitations in 2d. TODO del

Overview

RBL, as presented in the original paper [1], ensures convergence to the goal and provides sufficient conditions for achieving it. The problem involves individual control of N agents from their initial position $\mathbf{p}_i(0)$ toward a goal region, represented as circle. This goal region is denoted as $B(\mathbf{e}_i, \epsilon)$, where $\mathbf{e}_i$ is center and ϵ is radius of goal region. Each agent is knows of its current position $\mathbf{p}_i$, encumbrance δ_i, which determines safe space around agent. Additionally, each agent also knows the positions and encumbrances of its neighboring agents $\mathcal{N}_\mathbf{i}$, agent $j \in \mathcal{N}_\mathbf{i}$ if $||\mathbf{p}_i - \mathbf{p}_j|| \leq 2r_{s,i}$, where $r_{s,i}$ is denoted as half of the sensing radius of the i-th agent. For simplicity $r_{s,i}$ is considered to be same for all agents, therefore $r_{s,i} = r_s$.

TODO, write limitations and motivation for extension here.

Key principles

The core idea is to minimize the following cost funciton:

$$J_{cov}(\mathbf{p}) = \sum_{i=1}^{N} \int_{\mathcal{V}_i} \|\mathbf{q} - \mathbf{p}_i\|^2 \phi_i(\mathbf{q}) d\mathbf{q}, \tag{2.1}$$

where $\mathbf{p}_i$ is the position of agent i, $\mathcal{V}_i$ is the Voronoi cell of the i-th robot, $\|\mathbf{q} - \mathbf{p_i}\|^2$ is squared Euclidian distance between point in the mission space $\mathbf{q} \in \mathcal{Q}$ and agent's position p_i, and $\phi_i(\mathbf{q})$ is the weighting function.

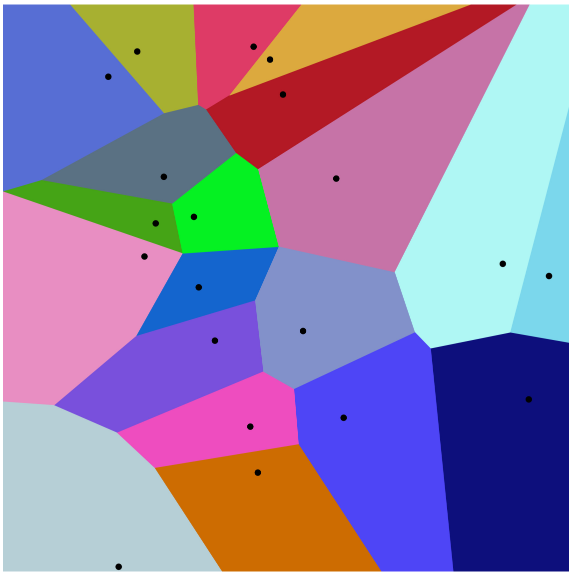
Voronoi cell is defined as:

$$\mathcal{V}_i = \{q \in \mathcal{Q} | \|\mathbf{q} - \mathbf{p}_i\| \leq \|q - \mathbf{p}_j\|, \forall j \neq i\} \tag{2.2}$$

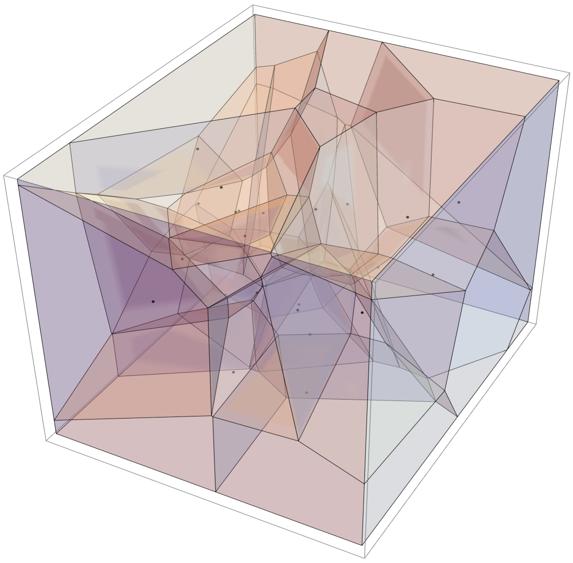
, for visual representation see Fig. 2.1a. However, this standard definition of Voronoi cells does not take into account the physical space occupied by the agents, or their encumbrances. To address this, a Modified Voronoi cell is introduced, which takes into account the encumbrances of agents. This modified version adjusts the boundaries of each Voronoi cell to account for the encumbrances of neighboring agents. The modified Voronoi cell definition is as follows:

$$\tilde{V}_i = \begin{cases} \{\mathbf{q} \in Q \mid \|\mathbf{q} - \mathbf{p}_i\| \leq \|\mathbf{q} - \mathbf{p}_j\|\}, & \text{if } \Delta_{ij} \leq \frac{\|\mathbf{p}_i - \mathbf{p}_j\|}{2} \\ \{\mathbf{q} \in Q \mid \|\mathbf{q} - \mathbf{p}_i\| \leq \|\mathbf{q} - \tilde{\mathbf{p}}_j\|\}, & \text{otherwise,} \end{cases} \tag{2.3}$$

$\forall j \in \mathcal{N}_i$, where $\Delta_{ij} = \delta_i + \delta_j$ and $\tilde{\mathbf{p}}_j = \mathbf{p}_j + 2(\Delta_{ij} - \frac{\|\mathbf{p}_i - \mathbf{p}_j\|}{2})\frac{\mathbf{p}_i - \mathbf{p}_j}{\|\mathbf{p}_i - \mathbf{p}_j\|}$.



(a) Euclidean Voronoi Diagram in 2D



(b) Euclidean Voronoi Diagram in 3D

Figure 2.1: (a) an example of 20 Voronoi cells in 2D [2] (b) 25 Voronoi cells in 3D [3]

Convergence to goal reagion $B(\mathbf{e}_i, \epsilon)$ depends on the choice of weighting function that assigns weights to points $\mathbf{q}$ in the mission space $\mathcal{Q}$. The weighting function $\phi_i(\mathbf{q})$ is defined as follows:

$$\phi_i(\mathbf{q}) = \exp\left(-\frac{\|\mathbf{q} - \bar{p}_i\|}{\beta_i}\right) \tag{7}$$

where

$$\dot{\beta}_i(A_i) = \begin{cases} -\beta_i & \text{if } \|\mathbf{c}_{A_i} - \mathbf{p}_i\| < d_1 \wedge \|\mathbf{c}_{A_i} - \mathbf{c}_{S_i}\| > d_2, \\ -(\beta_i - \beta_i^D) & \text{otherwise.} \end{cases} \tag{8}$$

$$\dot{\bar{p}}_i = \begin{cases} -(\bar{p}_i - R\mathbf{p}_i(\frac{\pi}{2} - \epsilon)\mathbf{e}_i) & \text{if } \|\mathbf{c}_{A_i} - \mathbf{p}_i\| < d_3 \wedge \|\mathbf{c}_{A_i} - \mathbf{c}_{S_i}\| > d_4, \\ -(\bar{p}_i - \mathbf{e}_i) & \text{otherwise,} \end{cases} \tag{2.4}$$

where

$$\bar{p}_i = \begin{cases} \mathbf{e}_i & \text{if } \|\mathbf{p}_i - \mathbf{c}_{A_i}\| > \|\mathbf{p}_i - \mathbf{c}_{A_i}\|, \\ R\mathbf{p}_i(\frac{\pi}{2} - \epsilon)\mathbf{e}_i & \text{otherwise.} \end{cases}$$

The position of the centroid $\mathbf{c}_{V_i}$ of a region V_i, weighted by the function $\phi_i(q)$, is used to guide the control. It is defined as:

$$\mathbf{c}_{V_i} = \frac{\int_{V_i} \mathbf{q}\phi_i(\mathbf{q})\, d\mathbf{q}}{\int_{V_i} \phi_i(\mathbf{q})\, d\mathbf{q}}, \tag{2.5}$$

where $\mathbf{q}$ represents the position vector, and $\phi_i(\mathbf{q})$ is a weighting function. The centroid $\mathbf{c}_{V_i}$ serves as the target for the control system, directing the system toward the weighted center of the region.

Applications in 2D

Limitations in 2D

2.2 Motivation for 3d Extension

Importance and challenges

2.3 Mathematical Describtion in 3D

modified voronoi partitioning, transition from planar 2d to spatial 3d, safety and convergence in 3d.

2.4 Simulation and Results Analysis

Describtion of simulation enviroment, used tools, Describtion of few simulation scenarios and result analysis.

2.5 Summary and Key Insights

Recap of modifications. Faced challenges and solutions applied

3 Environment Perception Using 3D LiDAR

3.1 Introduction

Motivation for using 3d lidar and challenges

3.2 3D LiDAR Sensor Model and Simulation Setup

Overview of LiDAR used. Simulation configuration

3.3 Object Detection and Approximation

Methods for extracting objects from LiDAR point clouds. Approximating detected objects with simple shapes. Handling noisy or incomplete data

3D Surface Approximation from Point Clouds ← This one seems promising. I would like to try this.

Alpha Shape: A Generalization of the Convex Hull

3.4 Integration with RBL Algorithm

Modifications to ensure safe navigation and how approximated objects influence Voronoi cells

3.5 Simulation Results

Example scenarios - in rviz with drone and also in real life - me holding a branch with leafs or something like that

3.6 Discussion

Limitations and possible improvements

3.7 Summary

4 Researching on possible enhancements of the algorithm for example using neural networks or learning-based techniques.

tune the paramaters of RBL to suit 3D

I will have to read something first, but what I found is that I could try object detection and classification using models like PointNet, PointNet++ or VoxelNet

Pointcloud denoising

Segmentation of LiDAR data. RangeNet++, SalsaNext to differentiate between terrain trees and free space

NN approaches for approximating point clouds with simple convex shapes:

CvxNet: Learnable Convex Decomposition by Deng et al. (CVPR 2020)

Label-Efficient Learning on Point Clouds using Approximate Convex Decompositions by Gadelha et al. (Arxiv 2020)

Region Segmentation via Deep Learning and Convex Optimization by Sonntag and Morgenshtern (Arxiv 2019)

Point Density-Aware Voxels for LiDAR 3D Object Detection

5 Conclusion

Summarize the achieved results. Can be similar as an abstract or an introduction, however, it should be written in past tense.

6 References

[1] M. Boldrer, A. Serra-Gomez, L. Lyons, V. Kratky, J. Alonso-Mora, and L. Ferranti, "Rule-based lloyd algorithm for multi-robot motion planning and control with safety and convergence guarantees," *arxiv*, 2024. [Online]. Available: https://arxiv.org/abs/2310.19511.

[2] B. Ertl, *3d voronoi mesh of 25 random points with 0.3 opacity and points*, Own work, licensed under CC BY-SA 4.0, Accessed: 2025-02-27, 2023. [Online]. Available: https://commons.wikimedia.org/w/index.php?curid=38534275.

[3] WalkingRadiance, *3d voronoi mesh of 25 random points with 0.3 opacity and points*, Own work, licensed under CC BY-SA 4.0, Accessed: 2025-02-27, 2023. [Online]. Available: https://commons.wikimedia.org/w/index.php?curid=127768757.

A Appendix A