

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



Vũ Đức Lương

**TÍNH TOÁN TỐI ƯU QUÁ TRÌNH CHUYỂN ĐỔI ĐỘI
HÌNH DRONE TRONG TRÌNH DIỄN ÁNH SÁNG**

ĐỒ ÁN TỐT NGHIỆP HỆ ĐẠI HỌC CHÍNH QUY

Ngành: Công nghệ hàng không vũ trụ

HÀ NỘI – 2025

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Vũ Đức Lương

TÍNH TOÁN TỐI ƯU QUÁ TRÌNH CHUYỂN ĐỔI ĐỘI
HÌNH DRONE TRONG TRÌNH DIỄN ÁNH SÁNG

ĐỒ ÁN TỐT NGHIỆP HỆ ĐẠI HỌC CHÍNH QUY

Ngành: Công nghệ hàng không vũ trụ

Cán bộ hướng dẫn: TS. Nguyễn Hoàng Quân

HÀ NỘI – 2025

TÓM TẮT

Tóm tắt: Trình diễn ánh sáng bằng drone là một hình thức biểu diễn trong đó hàng trăm hàng nghìn drone bay theo lộ trình đã được lập sẵn để tạo ra những hình dáng đội hình khác nhau trên bầu trời, tạo nên những hình ảnh và hiệu ứng thị giác ấn tượng. Công nghệ này ngày càng được ứng dụng trong các sự kiện lớn hay các lễ hội như một giải pháp thay thế cho pháo hoa truyền thống. Trong đồ án này, thuật toán Hungarian được sử dụng để tối ưu hóa việc gán nhiệm vụ với chi phí được tính toán dựa vào khoảng cách của các drone đến vị trí mục tiêu tương ứng từ các đội hình được thiết kế sẵn, nhằm tìm phương pháp phân công tối ưu với tổng quãng đường nhỏ nhất. Để đảm bảo quá trình chuyển đổi đội hình được mượt mà và an toàn, một thuật toán quy hoạch đường đi được kết hợp với thuật toán trường thế năng nhân tạo (Artificial Potential Field-APF) được kết hợp với thuật toán hungarian nhằm có thể tính toán quỹ đạo di chuyển của từng drone trong quá trình chuyển đổi đội hình. Thuật toán trường thế năng nhân tạo được cải tiến với trường đây được phân lớp để giảm dao động cho drone khi di chuyển, đồng thời các ràng buộc động học cũng được thêm vào để tăng độ ổn định và an toàn cho drone trong quá trình chuyển động. Bên cạnh đó thuật toán hoán đổi mục tiêu cũng được tích hợp nhằm giúp các drone có thể tránh được việc bị kẹt trong các điểm cục bộ trong quá trình di chuyển giúp các drone đều có thể hoàn thành được nhiệm vụ. Ngoài ra, đồ án cũng triển khai tăng tốc tính toán bằng GPU thông qua PyOpenCL nhằm giảm thời gian xử lý cho thuật toán quy hoạch đường đi 3D trong các mô phỏng chuyển đổi đội hình với số lượng drone lớn. Các kết quả mô phỏng về khoảng cách giữa các drone, vận tốc, gia tốc và thời gian bay, cho thấy phương pháp đề xuất hoạt động hiệu quả và có tính ứng dụng cao cho các đội hình drone quy mô lớn.

Từ khóa: Drone trình diễn ánh sáng, thuật toán Hungarian, quy hoạch đường đi, thuật toán trường thế năng nhân tạo, GPU, PyOpenCL

LỜI CAM ĐOAN

Tôi tên là Vũ Đức Lương, sinh viên lớp K66AE, Viện Công nghệ Hàng không Vũ trụ, Trường Đại học Công nghệ - Đại học Quốc gia Hà Nội.

Tôi xin cam đoan rằng đồ án tốt nghiệp với đề tài ”Tính toán tối ưu quá trình chuyển đổi đội hình drone trong trình diễn ánh sáng” là công trình nghiên cứu của tôi. Mọi sự giúp đỡ trong quá trình thực hiện đồ án đã được cảm ơn, các thông tin trích dẫn trong đồ án này đều được ghi rõ nguồn tham khảo theo đúng quy định. Các số liệu nghiên cứu và kết quả trình bày trong đồ án là trung thực.

Nếu có bất kỳ sai phạm nào, tôi xin hoàn toàn chịu trách nhiệm.

Hà Nội, ngày tháng năm 2025

Sinh viên

Vũ Đức Lương

LỜI CẢM ƠN

Trước hết, em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy giáo hướng dẫn là TS. Nguyễn Hoàng Quân và KS. Trần Đăng Huy đã tận tình hướng dẫn, giúp đỡ em trong quá trình thực hiện đồ án tốt nghiệp này. Bên cạnh đó em cũng gửi lời cảm ơn đến Viện Công nghệ Hàng không Vũ trụ và Trường Đại học Công nghệ đã cho em một môi trường học tập tốt với các cán bộ giảng viên có chuyên môn cao và tận tình trong công tác giảng dạy.

Em cũng xin lời cảm ơn đến gia đình và bạn bè đã luôn động viên, giúp đỡ em trong suốt quá trình học tập và thực hiện đồ án tốt nghiệp.

Mặc dù đã cố gắng hết sức, nhưng do hạn chế về kiến thức và thời gian, đồ án không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý từ thầy cô và các bạn trong hội đồng bảo vệ để hoàn thiện hơn trong những nghiên cứu sau này.

Mục lục

Chương 1. Tổng quan	1
1.1. Giới thiệu về drone trình diễn ánh sáng	1
1.2. Phần mềm điều hành và hệ thống điều khiển UAV bầy đàn trong drone trình diễn ánh sáng.....	2
1.3. Mục tiêu đề tài	7
Chương 2. Cơ sở lý thuyết thuật toán phân công nhiệm vụ và quy hoạch đường đi.....	8
2.1. Thuật toán phân công nhiệm vụ.....	8
2.1.1. Mô tả bài toán.	8
2.1.2. Thuật toán Hungarian	9
2.2. Thuật toán quy hoạch đường đi	12
2.2.1. Giới thiệu bài toán quy hoạch đường đi.....	12
2.2.2. Thuật toán trường thé năng nhân tạo	14
2.2.3. Thuật toán trường năng nhân tạo cải tiến cho trình diễn ánh sáng sử dụng drone.....	15
2.3. Khung chương trình tính toán điểm điều hướng được đề xuất.....	19
Chương 3. Lập trình song song GPU tăng tốc cho thuật toán quy hoạch đường đi.....	21
3.1. Đặt vấn đề	21
3.2. Giới thiệu về OpenCL và PyOpenCL	21
3.2.1. Lý do lựa chọn PyOpenCL cho đồ án.	22
3.2.2. Kiến trúc lập trình OpenCL.	22
3.3. Lập trình song song GPU cho khung chương trình đề xuất.	24
3.3.1. Lựa chọn thành phần song song hóa.	24
3.3.2. Triển khai thành phần tính toán lực tổng hợp trên GPU.	25
Chương 4. Thực hiện mô phỏng và phân tích kết quả.....	27
4.1. Kết quả về thời gian tính toán.	27
4.2. Kết quả chất lượng màn trình diễn.	28
4.2.1. Các thông số thuật toán ảnh hưởng đến chất lượng màn trình diễn.	28

4.2.2. So sánh với kết quả của phương pháp Sun [10].	35
Chương 5. Kết luận và hướng phát triển	37

Danh sách hình vẽ

1.1. Màn trình diễn ánh sáng của Damoda trong lễ kỉ niệm 50 năm giải phóng miền Nam thống nhất đất nước với 10500 drone [3].	1
1.2. Cấu tạo của một drone được sử dụng trong trình diễn ánh sáng [4].....	2
1.3. Luồng hoạt động của hệ thống [10].....	4
1.4. Hệ thống được triển khai bởi Sun cùng cộng sự [10].....	4
1.5. a) Các thành phần phần mềm của hệ thống UAV-LS, b) Thuật toán chuyển đổi đội hình Huang đề xuất [4].	5
1.6. Hệ thống UAV-LS được triển khai trong thực tế [4].	6
1.7. a) Kết quả phân công nhiệm vụ b) Kết quả quãng đường di chuyển của thuật toán quy hoạch đường đi.	7
2.1. Một robot được điều hướng đến đích và tránh vật cản nhờ thuật toán trường thế năng nhân tạo.....	15
2.2. Quãng đường drone bị dao động	17
2.3. Quãng đường drone được phân lớp	17
2.4. Minh họa hiện tượng hoán đổi vị trí	18
2.5. Khung chương trình đề xuất	20
3.1. Quy trình lập trình với PyOpenCL.....	23
3.2. So sánh thời gian thực thi công ma trận trên CPU và GPU	24
3.3. Khung chương trình tính toán điểm điều hướng với tăng tốc GPU	25
4.1. Kịch bản khảo sát	29
4.2. Khoảng cách nhỏ nhất của các drone trong quá trình chuyển đổi trong trường hợp 1	29
4.3. Vận tốc của các drone trong quá trình chuyển đổi trong trường hợp 1.....	30
4.4. Gia tốc của các drone trong quá trình chuyển đổi trong trường hợp 1	30
4.5. Khoảng cách nhỏ nhất của các drone trong quá trình chuyển đổi trong trường hợp 2	31
4.6. Vận tốc của các drone trong quá trình chuyển đổi trong trường hợp 2.....	32
4.7. Gia tốc của các drone trong quá trình chuyển đổi trong trường hợp 2	32
4.8. Khoảng cách nhỏ nhất của các drone trong quá trình chuyển đổi trong trường hợp 3	33

4.9. Vận tốc của các drone trong quá trình chuyển đổi trong trường hợp 3.....	34
4.10. Gia tốc của các drone trong quá trình chuyển đổi trong trường hợp 3.....	34
4.11. Kết quả của vận tốc của phương pháp APF [10].....	36
4.12. Kết quả của gia tốc phương pháp APF [10]	36

Danh sách bảng

1.1. Thông số kĩ thuật của drone trình diễn ánh sáng	2
2.1. Ma trận chi phí D.....	10
2.2. Xác định giá trị nhỏ nhất mỗi hàng	10
2.3. Ma trận sau khi được giảm hàng	11
2.4. Phủ các số 0 - Lần lặp 1	11
2.5. Ma trận kết quả sau các thao tác thuật toán-Lần lặp 1	11
2.6. Các phần tử 0 được lựa chọn	12
2.7. Kết quả phân công cuối cùng.....	12
4.1. So sánh thời gian tính toán trên CPU và GPU	27
4.2. Các thông số của buổi trình diễn	35

Các từ viết tắt

APF: Artificial Potential Field – Thuật toán trường thế năng nhân tạo.

GPU: Graphics Processing Unit – Bộ xử lý đồ họa.

CPU: Central Processing Unit – Bộ xử lý trung tâm.

UAV: Unmanned Aerial Vehicle – Máy bay không người lái.

PyOpenCL : Python OpenCL – Thư viện Python cho OpenCL.

GCS : Ground Control Station – Trạm điều khiển mặt đất.

GNSS-RTK: Global Navigation Satellite System – Real-Time Kinematic – Kỹ thuật định vị vệ tinh độ chính xác cao

GPS: Global Positioning Systems – Hệ thống định vị toàn cầu.

UAV-LS: Unmanned Aerial Vehicle - Light Show – Hệ thống trình diễn ánh sáng được trình bày bởi Huang [4].

ADI: Animation Design Interface – Giao diện thiết kế đội hình được đề xuất bởi Huang [4]

TAP: Task Assignment Problem – Bài toán phân công nhiệm vụ

Chương 1.

Tổng quan

1.1. Giới thiệu về drone trình diễn ánh sáng

Trong những năm gần đây, trình diễn ánh sáng bằng drone (drone light show) đã trở thành một xu hướng mới trong lĩnh vực giải trí và sự kiện. Công nghệ này sử dụng hàng trăm hàng nghìn UAV được lập trình để bay theo các đội hình và tạo ra những hình ảnh, hiệu ứng ánh sáng ấn tượng trên bầu trời. Nhờ vào khả năng tái sử dụng, không gây khó bụi và có tính linh hoạt cao cũng như mức độ an toàn tốt hơn mà việc sử dụng drone trong các màn trình diễn ánh sáng ngày càng được sử dụng rộng rãi đặc biệt trong những sự kiện lớn thay thế dần cho pháo hoa truyền thống.

Với sự phát triển nhanh chóng của UAV và công nghệ điều khiển tự động đã ghi nhận nhiều màn trình diễn ánh sáng với số lượng drone lớn được thực hiện thành công bởi các công ty như Intel[1] hay Ehang[2] đặc biệt trong lễ kỉ niệm 50 năm giải phóng miền Nam thống nhất đất nước công ty Damoda đã đạt kỉ lục Guiness với một màn trình diễn ánh sáng sử dụng 10500 drone [3], từ những thành công này có thể thấy được drone light show là một công nghệ mới nổi nhưng đầy tiềm năng phát triển trong tương lai. Do đó đã có nhiều nghiên cứu về điều khiển bầy đàn, truyền thông giữa các drone, thuật toán quy hoạch đường đi, thiết kế phần cứng và phần mềm nhằm hướng tới các buổi trình diễn ánh sáng sáng tạo hơn, đẹp mắt hơn và tiết kiệm năng lượng hơn. Hình 1.1 được lấy từ buổi trình diễn ánh sáng do Damoda trình diễn trong lễ kỉ niệm 50 năm giải phóng miền Nam thống nhất đất nước.



Hình 1.1. Màn trình diễn ánh sáng của Damoda trong lễ kỉ niệm 50 năm giải phóng miền Nam thống nhất đất nước với 10500 drone [3].

Sơ lược về phần cứng của drone sử dụng trong trình diễn ánh sáng, thông thường các drone bốn cánh quạt (quadrotor) là lựa chọn phổ biến do có thể phát triển với giá thành thấp và cấu tạo linh hoạt. Trong nghiên cứu của Huang và công sự thì quadrotor có cấu tạo từ 3 phần như hình 1.1 với trung tâm điều khiển có dạng elipse, một khung sợi cánh quạt và bốn cánh quạt. Trung tâm điều khiển bay chứa các bộ điều khiển tích hợp, mô đun truyền thông không dây hai kênh, một bộ định vị, pin lithium có thể sạc lại được và mô đun led được gắn ở vị trí dưới của drone. Bốn cụm động cơ gồm cánh quạt của quadrotor bao gồm động cơ không chổi than và một cánh quạt có ba lá cánh, khung sợi carbon nhằm nối liền trung tâm điều khiển và bốn cụm động cơ cánh quạt [4]. Một số thông số kỹ thuật cơ bản của drone đã được công bố bởi các hãng như Collmot [5] hay Damoda [6] được trình bày ở bảng 1.1, qua đó có thể thấy rằng các drone được sử dụng của Damoda có chất lượng hàng đầu hiện nay điều này giúp công ty này có thể thực hiện được các màn trình diễn với quy mô không tưởng. Về phần mềm cũng như quá trình điều khiển bay sẽ được trình bày chi tiết trong mục 1.2.



Hình 1.2. Cấu tạo của một drone được sử dụng trong trình diễn ánh sáng [4].

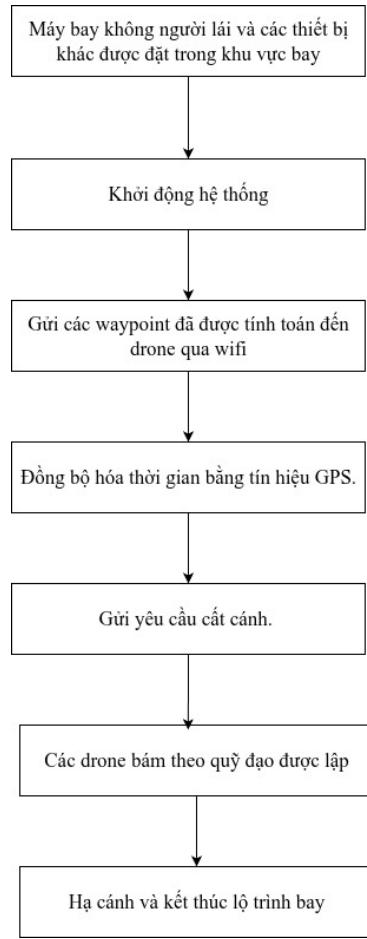
Bảng 1.1. Thông số kỹ thuật của drone trình diễn ánh sáng

Thông số	Collmot[5]	Light show Drone L3 (Damoda) [6]
Loại	Quadrotor	Quadrotor
Trọng lượng	< 2 kg	530 g
Kích thước	Đường kính < 100 cm, cao ≈ 30 cm	Đường kính 320 mm, cao 115 mm
Thời gian bay	10–12 phút	20–25 phút
Tốc độ tối đa	6 m/s ngang, 2 m/s dọc	10 m/s
Độ cao tối đa	120 m	120 m
Khoảng cách tối thiểu	7 m	1.4 m

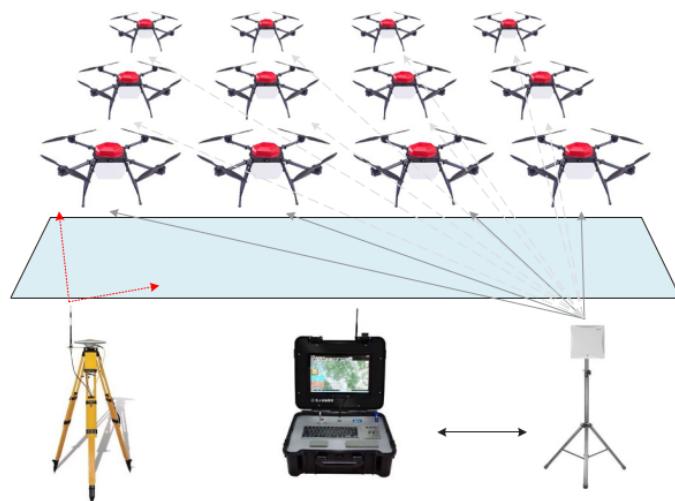
1.2. Phần mềm điều hành và hệ thống điều khiển UAV bảy đàn trong drone trình diễn ánh sáng.

Có nhiều phương pháp điều khiển đội hình UAV, thường có một trạm điều khiển có thể là một máy tính hiệu năng cao hoặc các thiết bị di động có trách nhiệm quản lý và điều khiển các UAV [7]. Dựa vào ảnh hưởng của trạm mặt đất, có thể phân loại các phương pháp điều khiển như phương pháp thủ công (human in the loop) với các drone được điều khiển hoàn toàn bằng trạm mặt đất trong suốt quá trình buổi diễn, phương pháp bán tự động (semi-autonomous) các bộ điều khiển bay trên drone sẽ phối hợp điều khiển với các bộ điều khiển bay trên drone sẽ phối hợp điều khiển với trạm mặt đất, trạm mặt đất có chức năng gửi các điểm điều hướng (waypoint) đến các drone, bộ điều khiển bay của từng drone sẽ giúp duy trì trình ổn định bay cũng như tránh va chạm, phương pháp tự động (autonomous) các drone sẽ tự đưa ra kế hoạch bay và tránh vật cản mà không phụ thuộc nhiều vào chạm mặt đất [8]. Trong các buổi trình diễn ánh sáng bằng drone, phương pháp điều khiển bán tự động thường được sử dụng phổ biến nhất do phương pháp điều khiển tự động thường yêu cầu cảm biến cảm biến và khả năng tính toán của bộ điều khiển rất phức tạp ở từng drone trong khi đó phương pháp thủ công lại rất rủi ro không phù hợp với các màn trình diễn quy mô lớn vì dễ gây nguy cơ va chạm do phụ thuộc vào đường truyền của UAV với trạm mặt đất cũng như kỹ thuật của phi công điều khiển UAV [4].

Hiện nay, hầu hết các phần mềm của các công ty đều được phát triển công phu tiêu biểu nhưng thường được thương mại hóa với chi phí rất cao và không công khai công nghệ lõi [9]. Mặc dù vậy có một số nhóm nghiên cứu có công bố khá chi tiết về kiến trúc phần mềm và hệ thống điều khiển trong hệ thống thực tế khi triển khai các màn trình diễn ánh sáng. Như trong nghiên cứu của Sun cùng cộng sự, các đội hình sẽ được thiết kế sẵn qua các phần mềm đồ họa như Blender, thuật toán quy hoạch đường đi được đề xuất nhằm có thể xuất ra các waypoint cho quá trình chuyển đổi đội hình chi tiết về thuật toán này sẽ được trình bày cụ thể trong chương hai, các waypoint sau khi được tính toán sẽ được gửi đến các drone qua wifi bộ điều khiển bay trên drone sẽ giúp các drone bám theo các waypoint này, hệ thống định vị GNSS-RTK được sử dụng để đảm bảo sai số định vị chỉ ở mức cm [10]. Quy trình thực hiện được mô tả qua hình 1.3, cấu trúc hệ thống được thể hiện qua hình 1.4. Lưu ý, các mô phỏng này được thực hiện nhằm mục đích khảo sát tốc độ tính toán của thuật toán quy hoạch đường đi nên các tham số trong thuật toán được điều chỉnh để có thể các drone nhanh đến đích hơn từ đó giảm số vòng lặp tính toán chưa quan tâm đến tính an toàn trong bay thực tế như tăng gia tốc tối đa, vận tốc tối đa, giảm bán kính trường đầy hay tăng hệ số hấp dẫn.

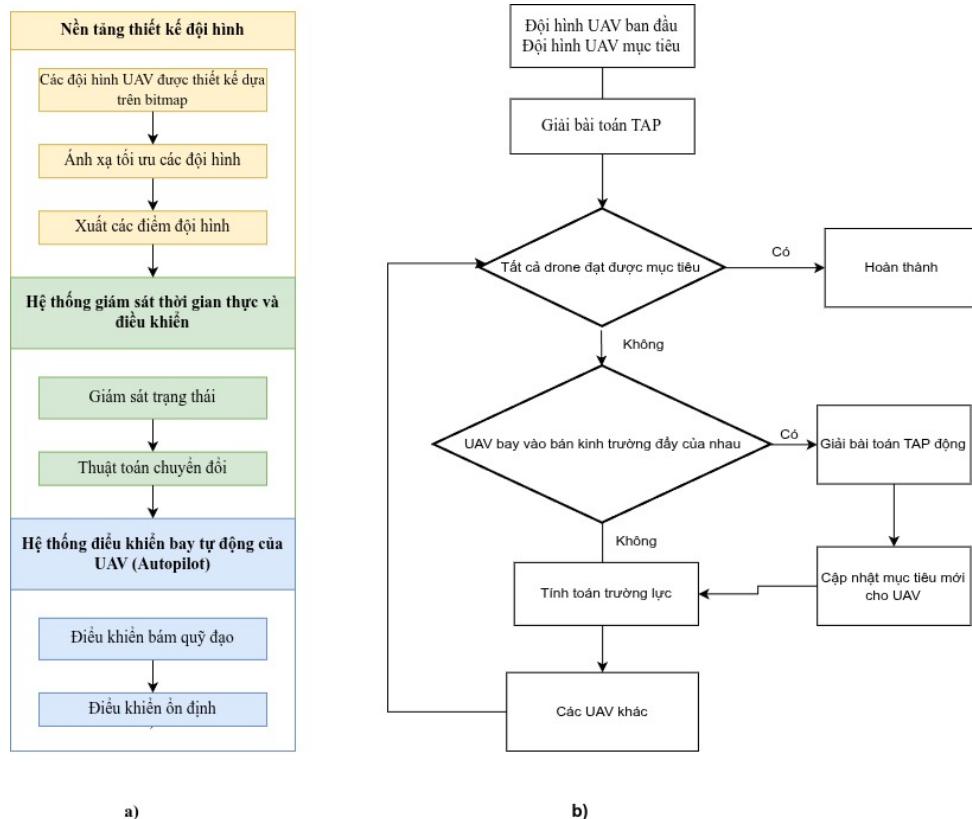


Hình 1.3. Luồng hoạt thông của hệ thống [10].



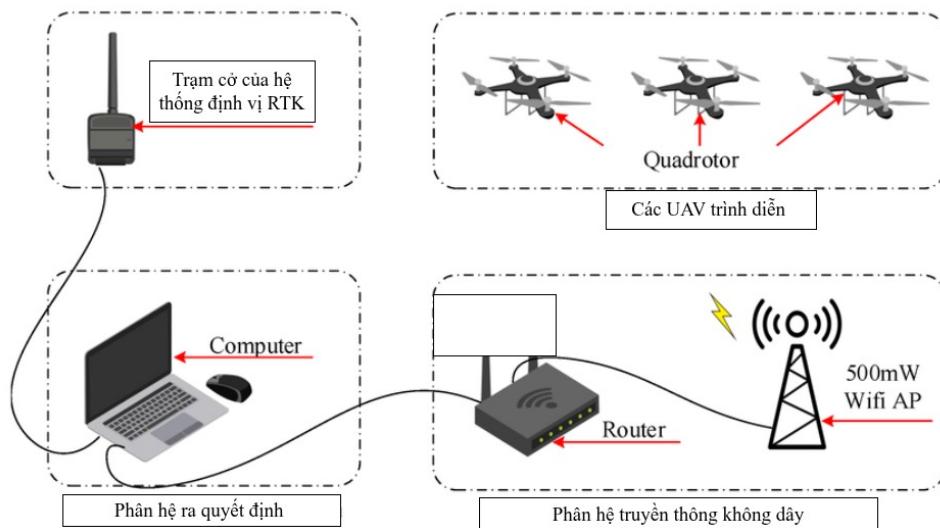
Hình 1.4. Hệ thống được triển khai bởi Sun cùng cộng sự [10].

Trong nghiên cứu của Huang cùng cộng sự hệ thống bay được triển khai trong thực tế được trình bày chi tiết hơn cả về phần cứng và phần mềm, một hệ thống UAV - LS được giới thiệu với một thuật toán tính toán chuyển đổi đội hình động (online), một nền tảng giúp thiết kế các đội hình drone (ADI) và ứng dụng giám sát thời gian thực như hình 1.5. Hệ thống ADI cho phép người dùng thiết kế đội hình UAV trực quan bằng cách vẽ trên bitmap, trong đó mỗi chấm trên bitmap xác định tọa độ (kinh độ, vĩ độ) của một UAV. Ngay sau khi hoàn thành việc vẽ đội hình, thuật toán Hungarian giải quyết bài toán phân công nhiệm vụ sẽ được kích hoạt để giải bài toán tối ưu ghép cặp các UAV từ vị trí ban đầu đến vị trí mục tiêu trong đội hình mới. Cuối cùng, các tọa độ này được hệ thống tự động chuyển đổi thành các điểm đường bay để sẵn sàng cho việc điều khiển chuyến bay. Thuật toán chuyển đổi đội hình được đề xuất với việc giải bài toán TAP theo thuật toán Hungarian theo thiết lập có điều kiện cũng như thuật toán APP dùng để tránh va chạm cũng như đưa drone đến đích. Điều kiện để thực hiện thuật toán Hungarian động là khi các drone bay vào trường đầy của nhau, chỉ thực hiện giải TAP cho cụm các UAV này cải tiến này đã giúp làm cho quãng đường UAV phải dịch chuyển để đạt được đội hình mục tiêu được giảm đi đáng kể chi tiết thuật toán được trình bày như trong hình 1.5 [4].



Hình 1.5. a) Các thành phần phần mềm của hệ thống UAV-LS, b) Thuật toán chuyển đổi đội hình Huang đề xuất [4].

Như trong hình 1.6, hệ thống UAV-LS được triển khai trong thực tế bao gồm một phân hệ ra quyết định, hệ thống định vị GNSS-RTK được sử dụng, phân hệ truyền thông không dây, và các UAV trình diễn như hình 1.2. Các thông tin từ cảm biến sẽ được gửi từ UAV về trạm mặt đất qua phân hệ truyền thông không dây theo chu kỳ. Các thông tin này bao gồm lat, lon thu được từ GNSS - RTK, cũng như độ cao của UAV được thu thập qua cảm biến áp suất. UAV và GCS là hai nền tảng khác nhau, chạy các hệ điều hành khác nhau nên việc trao đổi thông tin được thực hiện qua giao thức MAVLINK. Ở trạm mặt đất, một chuỗi các quyết định được đưa ra và các quỹ đạo được thiết lập qua thuật toán như hình 1.5, sau đó gửi đến drone qua truyền thông không dây. Bộ điều khiển tự động trên UAV điều chỉnh chế độ di chuyển của UAV để bám theo các quỹ đạo tham chiếu.



Hình 1.6. Hệ thống UAV-LS được triển khai trong thực tế [4].

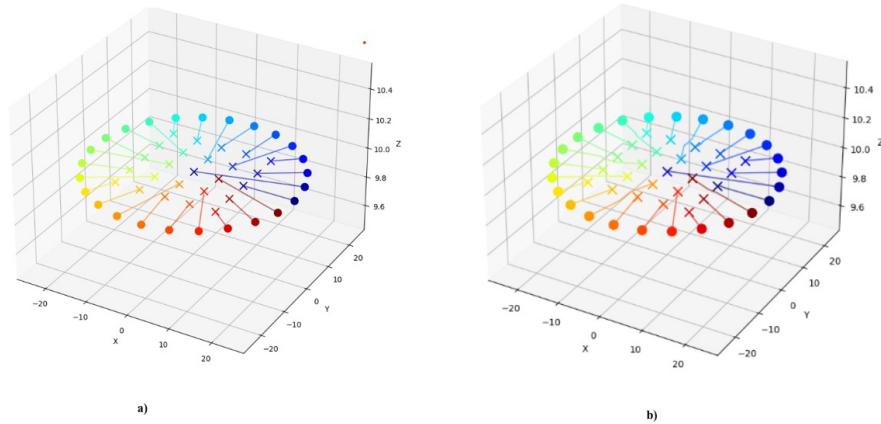
Mặc dù Huang cùng cộng sự đã nêu chi tiết về hệ thống họ triển khai cả về phần cứng và phần mềm, nhưng còn nhiều mặt hạn chế như khi triển khai với một số lượng drone lớn thì việc các drone bay vào bán kính trường đầy của nhau sẽ xảy ra với tần suất rất cao khiến bài toán TAP bị giải liên tục rất dễ gây ra quá tải cho GCS cũng như quỹ đạo sẽ bị thay đổi liên tục. Mặt khác việc giới hạn tốc độ cũng không được đề cập có thể làm cho các UAV bị mất ổn định giảm khả năng bám quỹ đạo gây mất an toàn cho buổi trình diễn. Thực tế nhóm nghiên cứu này chỉ thực hiện bay với 6 UAV nếu tăng số lượng đến hàng trăm UAV thì rất khó có thể triển khai được hệ thống này. Phương pháp của Sun đã triển khai một màn trình diễn với 500 drone cho thấy phương pháp này có triển vọng hoàn thành các màn trình diễn ánh sáng với số lượng drone lớn [10]. Các màn trình diễn ánh sáng thường được triển khai trong không gian trống đảm bảo không có vật cản nên việc thực hiện thuật toán quy hoạch đường đi trước để có được toàn bộ các waypoints cho chuyến động của drone rồi gửi đến các bộ điều khiển bay là một giải

pháp có thể chấp nhận được. Vì thế, phương pháp thực hiện như nghiên cứu [10] được lựa chọn để phát triển đồ án tốt nghiệp này.

1.3. Mục tiêu đề tài

Như đã đề cập, trong nghiên cứu [10] Sun cùng cộng sự đã triển khai thành công một màn trình diễn với 500 UAV, nghiên cứu đã đề xuất một thuật toán quy hoạch đường đi dựa trên phương pháp trường thế năng nhân tạo với các cải tiến như phân lớp trường đầy, thêm các ràng buộc về động học hay thêm một thuật toán hoán đổi đường đi giúp đảm bảo các drone đều có thể đến đích. Các nghiên cứu gần đây thì thuật toán phân công nhiệm vụ bay thường kết hợp với thuật toán quy hoạch đường đi để giúp các buổi trình diễn hiệu quả và tiết kiệm chi phí hơn [4], [11], [12].

Về thuật toán phân công nhiệm vụ , Nar cùng cộng sự đã đánh giá nhiều thuật toán tối ưu như giải thuật di truyền, nhánh cận, Hungarian được áp dụng cho bài toán tối ưu tổng chi phí khi gán một drone vào một waypoint kết quả cho thấy thuật toán Hungarian hiệu quả nhất kể cả về thời gian tính toán lẫn kết quả tối ưu tìm được [13]. Trong đồ án này, thuật toán phân công nhiệm vụ [13] sẽ được kết hợp với thuật toán lập quy hoạch đường đi như hình 1.7 nhằm giải quyết đồng thời hai vấn đề lớn trong trình diễn ánh sáng: (i) đảm bảo quãng đường di chuyển tổng thể của UAV là tối ưu, (ii) duy trì sự an toàn bay với các đường đi mượt, tránh được va chạm trong quá trình biến đổi đội hình. Ngoài ra, đồ án cũng triển khai tăng tốc tính toán bằng GPU thông qua PyOpenCL nhằm giảm thời gian xử lý cho thuật toán quy hoạch đường đi 3D trong các mô phỏng chuyển đổi đội hình với số lượng drone lớn.



Hình 1.7. a) Kết quả phân công nhiệm vụ b) Kết quả quãng đường di chuyển của thuật toán quy hoạch đường đi.

Chương 2.

Cơ sở lý thuyết thuật toán phân công nhiệm vụ và quy hoạch đường đi.

2.1. Thuật toán phân công nhiệm vụ

2.1.1. Mô tả bài toán.

Theo Chen [11], trong không gian 3 D mỗi drone được xem như là một vật rắn vị trí của drone được xem như là $\mathbf{p}_i = (x_i, y_i, z_i)$ vị trí bắt đầu, vị trí ở thời điểm t, vị trí mục tiêu được giả sử là $\mathbf{p}_i^s, \mathbf{p}_i^t, \mathbf{p}_i^e$. Các kịch bản trình diễn được xem xét trong nghiên cứu này thì sẽ bao gồm n UAV và n mục tiêu, do đó đội hình drone có vị trí ban đầu, vị trí thời điểm t và vị trí mục tiêu được thể hiện như công thức (2.1), (2.2), (2.3).

$$\mathbf{P}_S = [\mathbf{p}_1^s, \mathbf{p}_2^s, \mathbf{p}_3^s, \dots, \dots, \mathbf{p}_n^s - 1, \mathbf{p}_n^s] \quad (2.1)$$

$$\mathbf{P}_t = [\mathbf{p}_1^t, \mathbf{p}_2^t, \mathbf{p}_3^t, \dots, \dots, \mathbf{p}_n^t - 1, \mathbf{p}_n^t] \quad (2.2)$$

$$\mathbf{P}_e = [\mathbf{p}_1^e, \mathbf{p}_2^e, \mathbf{p}_3^e, \dots, \dots, \mathbf{p}_n^e - 1, \mathbf{p}_n^e] \quad (2.3)$$

Ma trận phân công nhiệm vụ S.

$$\mathbf{S} = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,n-1} & s_{1,n} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,n-1} & s_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ s_{n-1,1} & s_{n-1,2} & \cdots & s_{n-1,n-1} & s_{n-1,n} \\ s_{n,1} & s_{n,2} & \cdots & s_{n,n-1} & s_{n,n} \end{bmatrix} \quad (2.4)$$

Khi nhiệm vụ i được phân công cho drone j thì $s_{i,j} = 1$ ngược lại thì $s_{i,j} = 0$, mục tiêu của bài toán phân công đội hình là nhằm tối thiểu hóa tổng chi phí phân công và mỗi drone i chỉ được phân công cho một nhiệm vụ j. Do đó có thể xây dựng được hàm mục tiêu như công thức (2.5) và các ràng buộc cho bài toán phân công nhiệm vụ được xác định như công thức (2.6). $d_{i,j}$ trong công thức (2.5) là khoảng cách từ drone i đến mục tiêu j.

$$\min \sum_{i=1}^n \sum_{j=1}^n s_{i,j} d_{i,j} \quad (2.5)$$

$$\begin{cases} \sum_{j=1}^n s_{i,j} = 1, & 1 \leq i \leq n \\ \sum_{i=1}^n s_{i,j} = 1, & 1 \leq j \leq n \end{cases} \quad (2.6)$$

2.1.2. Thuật toán Hungarian

Thuật toán Hungarian là thuật toán được phát triển để giải quyết cho các bài toán phân công nhiệm vụ trong thời gian đa thức với mục tiêu là tối thiểu hóa tổng chi phí phân công là nhỏ nhất và mỗi nhiệm vụ chỉ được gán cho một đối tượng với độ phức tạp $O(n^3)$ [13]. Ứng dụng thuật toán Hungarian để giải quyết bài toán phân công nhiệm vụ bay sẽ được trình bày như sau.

Thuật toán 1 Áp dung thuật toán Hungarian để giải bài toán phân công nhiệm vụ bay

Đầu vào: Mảng của vị trí bắt đầu của drone, mảng vị trí mục tiêu trên đội hình được thiết kế

Đầu ra: Ánh xạ tối ưu giữa drone và mục tiêu π

1. Tạo ma trận $n * n$ khoảng cách Euclidean, \mathbf{D} với mỗi phần tử $D_{i,j}$ đại diện cho khoảng cách từ drone i và mục tiêu j như công thức 2.7

$$d_{i,j} = \|\mathbf{p}_j^e - \mathbf{p}_i^s\| \quad (2.7)$$

2. $\forall \text{row}(i) \in \mathbf{D}$
3. $\text{row}_{min} = \min(\mathbf{D}_{i,:})$
4. $\forall d \in \mathbf{D}_{i,:}, d = \text{row}_{min} - d$ thu được ít nhất một số 0 mỗi hàng
5. $\forall \text{col}(j) \in \mathbf{D}$
6. $\text{col}_{min} = \min(\mathbf{D}_{:,j})$
7. $\forall d \in \mathbf{D}_{:,j}, d = \text{col}_{min} - d$ thu được ít nhất một số 0 mỗi cột
8. Đặt l_{min} là số hàng nhỏ nhất bao phủ tất cả các số 0 trong D
9. if $l_{min} < n$
 10. Đặt $d_{ij,uc}$ là phân tử chưa bị bao phủ bởi bất kì đường thẳng nào.
 11. Đặt $d_{ij,int}$ là phân tử giao nhau giữa hai đường thẳng.
 12. $d_{ij,uc} = d_{ij,uc} - d_{min}$
 13. $d_{ij,int} = d_{ij,int} - d_{min}$

14. Return 12

15. if $l_{min} = n$ then

16. $M = \{di : wj\}$ sao cho $C_{ij} = 0$ và chỉ xuất hiện duy nhất một lần

$\forall row_i, \forall col_j$

Giá trị tối ưu cần tìm

Thuật toán Hungarian được áp dụng có thể được giải thích bằng minh họa số sau: Đầu vào $\mathbf{p}_i^s = [(5, 0, 20), (10, 0, 20), (15, 0, 20), (10, 0, 15), (10, 0, 10)]$, $\mathbf{p}_i^e = [(10, 0, 25), (20, 0, 18), (16, 0, 5), (4, 0, 5), (0, 0, 18)]$

(Dòng 1) Tạo ma trận chi phí như bảng 2.1.

Bảng 2.1. Ma trận chi phí D

	\mathbf{p}_1^s	\mathbf{p}_2^s	\mathbf{p}_3^s	\mathbf{p}_4^s	\mathbf{p}_5^s
\mathbf{p}_1^e	7.1	15.1	18.6	15.0	5.4
\mathbf{p}_2^e	5.0	10.2	16.1	16.1	10.2
\mathbf{p}_3^e	7.1	5.4	15.0	18.6	15.1
\mathbf{p}_4^e	10.0	10.4	11.7	11.7	10.4
\mathbf{p}_5^e	15.0	12.8	7.8	7.8	12.8

(Dòng 2 - 4) Giảm hàng: giá trị nhỏ nhất của mỗi hàng từ bảng 2.1 được đánh tô đỏ như được thấy trong bảng 2.2 các giá trị này được trừ khỏi từng phần của hàng đó để đảm bảo có ít nhất một phần tử 0 trong mỗi hàng như bảng 2.3.

Bảng 2.2. Xác định giá trị nhỏ nhất mỗi hàng

	\mathbf{p}_1^s	\mathbf{p}_2^s	\mathbf{p}_3^s	\mathbf{p}_4^s	\mathbf{p}_5^s
\mathbf{p}_1^e	7.1	15.1	18.6	15.0	5.4
\mathbf{p}_2^e	5.0	10.2	16.1	16.1	10.2
\mathbf{p}_3^e	7.1	5.4	15.0	18.6	15.1
\mathbf{p}_4^e	10.0	10.4	11.7	11.7	10.4
\mathbf{p}_5^e	15.0	12.8	7.8	7.8	12.8

Bảng 2.3. Ma trận sau khi được giảm hàng

	\mathbf{p}_1^s	\mathbf{p}_2^s	\mathbf{p}_3^s	\mathbf{p}_4^s	\mathbf{p}_5^s
\mathbf{p}_1^e	1.7	9.7	13.2	9.6	0
\mathbf{p}_2^e	0	5.2	11.1	11.1	5.2
\mathbf{p}_3^e	1.7	0	9.6	13.2	9.7
\mathbf{p}_4^e	0	0.4	1.7	1.7	0.4
\mathbf{p}_5^e	7.2	5.0	0	0	5.0

(Dòng 5 - 7) Giảm cột : Xác định giá trị nhỏ nhất ở mỗi cột các giá trị này được trừ khỏi từng phần của cột đó để đảm bảo có ít nhất một phần tử 0 trong mỗi cột. Trong bước này có thể thấy mỗi cột đều có ít nhất một giá trị 0, nên kết quả thu được vẫn như bước giảm hàng.

(Dòng 8 - 14) Tìm số đường nhỏ nhất có thể bao phủ hết tất cả các giá trị 0. Như có thể thấy trong bảng 6, tất cả các số 0 có thể bị bao phủ bằng các đường thẳng được vẽ ở cột 1, cột 2, cột 5 và hàng 5. Có thể thấy số đường cần có $l_{min} = 4$ nhỏ hơn số bậc của ma trận do đó kết quả tìm được chưa được tối ưu. Chọn giá trị nhỏ nhất từ tất cả các giá trị chưa được bao phủ $d_{ij,uc}$ trong trường hợp này giá trị tìm được là 1.7. Chọn $d_{ij,int}$ là những giá trị được giao giữa hai đường thẳng những giá trị này được bôi đỏ ở bảng 2.4. $d_{ij,uc}$ được trừ khỏi tất cả các ma trận chưa bao phủ và được thêm vào tất cả các giá trị cắt $d_{ij,int}$ kết quả thu được như ở bảng 2.5. Trong bảng 2.5 có thể thấy rằng các giá trị cần ít nhất 5 đường thẳng mới có thể bao phủ được hết hàng 4, hàng 5, cột 1, cột 2, cột 5.

Bảng 2.4. Phủ các số 0 - Lần lặp 1

	\mathbf{p}_1^s	\mathbf{p}_2^s	\mathbf{p}_3^s	\mathbf{p}_4^s	\mathbf{p}_5^s
\mathbf{p}_1^e	1.7	9.7	13.2	9.6	0
\mathbf{p}_2^e	0	5.2	11.1	11.1	5.2
\mathbf{p}_3^e	1.7	0	9.6	13.2	9.7
\mathbf{p}_4^e	0	0.4	1.7	1.7	0.4
\mathbf{p}_5^e	7.2	5.0	0	0	5.0

Bảng 2.5. Ma trận kết quả sau các thao tác thuật toán-Lần lặp 1

	\mathbf{p}_1^s	\mathbf{p}_2^s	\mathbf{p}_3^s	\mathbf{p}_4^s	\mathbf{p}_5^s
\mathbf{p}_1^e	1.7	9.7	11.5	9.6	0
\mathbf{p}_2^e	0	5.2	9.4	9.4	5.2
\mathbf{p}_3^e	1.7	0	7.9	13.2	9.7
\mathbf{p}_4^e	0	0.4	0	0	0.4
\mathbf{p}_5^e	8.9	6.7	0	0	6.7

(Dòng 15-16) Kết quả phân công cuối cùng: Chọn các phần tử bằng không từ ma trận kết quả sao cho mỗi hàng và mỗi cột chỉ được chọn một lần. Như thể hiện trong bảng 2.6 các phần tử bằng không được chọn được tô màu và sẽ được sử dụng để gán drone cho các điểm đường đi tương ứng. Chỉ số (i,j) của các phần tử 0 này sẽ đại diện cho việc phân công drone thứ i vào waypoint thứ j, . Kết quả phân công cuối cùng được hiển thị trong bảng 2.7.

Bảng 2.6. Các phần tử 0 được lựa chọn

	\mathbf{p}_1^s	\mathbf{p}_2^s	\mathbf{p}_3^s	\mathbf{p}_4^s	\mathbf{p}_5^s
\mathbf{p}_1^e	1.7	9.7	11.5	9.6	0
\mathbf{p}_2^e	0	5.2	9.4	9.4	5.2
\mathbf{p}_3^e	1.7	0	7.9	13.2	9.7
\mathbf{p}_4^e	0	0.4	0	0	0.4
\mathbf{p}_5^e	8.9	6.7	0	0	6.7

Bảng 2.7. Kết quả phân công cuối cùng

Drones	Mục tiêu	Chi phí
\mathbf{p}_1^s	\mathbf{p}_5^e	5.4
\mathbf{p}_2^s	\mathbf{p}_1^e	5.0
\mathbf{p}_3^s	\mathbf{p}_2^e	5.4
\mathbf{p}_4^s	\mathbf{p}_3^e	11.7
\mathbf{p}_5^s	\mathbf{p}_4^e	7.8
Tổng chi phí		35.4

2.2. Thuật toán quy hoạch đường đi

2.2.1. Giới thiệu bài toán quy hoạch đường đi

Quy hoạch đường đi (Path Planning) là một trong những bài toán quan trọng của robot di động và các hệ thống tự hành. Mục tiêu của quy hoạch đường đi là tìm ra một quỹ đạo khả thi, an toàn và tối ưu để một tác tử (agent) di chuyển từ vị trí ban đầu đến vị trí mục tiêu trong không gian hoạt động, đồng thời tránh va chạm với các vật cản hoặc các tác tử khác trong môi trường. Trong các hệ thống đa UAV như trình diễn ánh sáng, bài toán này có mức độ phức tạp rất cao do số lượng lớn UAV hoạt động đồng thời trong không gian hạn chế, dẫn đến mật độ tương tác lớn và yêu cầu an toàn nghiêm ngặt. Trong bối cảnh drone light show, mỗi UAV được gán một vị trí mục tiêu thông qua thuật toán phân công nhiệm vụ, nhưng để đảm bảo an toàn và đồng bộ thì mỗi UAV vẫn cần thực hiện quy hoạch đường đi chi tiết để tránh va chạm với các UAV khác trong suốt quá trình chuyển đổi đội hình. Vì vậy, quy hoạch đường đi không chỉ nhằm tối ưu quãng đường

mà còn phải đảm bảo các yêu cầu quan trọng như khoảng cách an toàn, tính khả thi động học gia tốc và vận tốc của drone phải ở ngưỡng yêu cầu, đường đi phải mượt mà hạn chế dao động hoặc thay đổi hướng đột ngột để đảm bảo UAV bám quỹ đạo tốt, không bị mắc kẹt trong quá trình chuyển động đảm bảo tất cả drone đều đến được vị trí mục tiêu trên đội hình được thiết kế. Có hai phương thức để giải bài toán quy hoạch đường đi online và offline được phân loại dựa trên sự biết trước của môi trường robot hoạt động [14].

Ở chế độ offline, tác tử có đầy đủ thông tin về môi trường bao gồm bản đồ, vật cản, vị trí bắt đầu và vị trí mục tiêu. Với dữ liệu toàn cục này, nhiệm vụ của quy hoạch đường đi offline là kết nối vị trí ban đầu tới vị trí mục tiêu bằng một quỹ đạo tối ưu. Robot chỉ cần thực hiện theo quỹ đạo đã được tính toán mà không cần cập nhật thêm trong quá trình bay. Phương pháp này đặc biệt phù hợp với môi trường tĩnh hoặc thay đổi rất chậm và cho phép tìm được đường đi tối ưu toàn cục về độ dài, an toàn, thời gian hoặc năng lượng. Ngược lại, quy hoạch đường đi online là quá trình mà trong đó quỹ đạo liên tục được cập nhật song song với quá trình UAV di chuyển và cảm nhận môi trường. Phương pháp này cần thiết trong môi trường động hoặc khi robot không biết đầy đủ bản đồ trước đó. Các thay đổi như vật cản mới, sai số vị trí hoặc chuyển động của UAV khác đều phải được xem xét và đưa vào quá trình tính toán. Tuy nhiên, do chỉ có thông tin cục bộ, quy hoạch đường đi online thường tiêu tốn nhiều thời gian tính toán hơn, yêu cầu cảm biến phức tạp, và không đảm bảo tìm được quỹ đạo tối ưu toàn cục. Trong một số trường hợp, robot có thể thậm chí không thể đến được mục tiêu, nhưng dù vậy nó vẫn là phương pháp bắt buộc trong các môi trường phức tạp và thay đổi [14].

Xuất phát từ đặc thù của bài toán trình diễn ánh sáng drone với một môi trường có mật độ drone dày đặc, động nhưng phần lớn đã được biết trước và có thể lập kế hoạch offline – việc lựa chọn một phương pháp quy hoạch đường đi cần cân nhắc các yếu tố then chốt sau: khả năng mở rộng để xử lý số lượng drone lớn, đảm bảo tính an toàn bằng cách duy trì khoảng cách, tạo ra quỹ đạo mượt mà phù hợp với động học của drone, và quan trọng là thời gian tính toán phải chấp nhận được để áp dụng trong thực tế. Thuật toán Trường Thé Năng Nhân Tạo được lựa chọn cho đồ án này nhờ những ưu điểm nổi bật phù hợp với yêu cầu đặt ra:

- Tính phản ứng và an toàn trong môi trường động: Cơ chế hoạt động của APF dựa trên việc mỗi drone chịu tác động của lực hút từ mục tiêu và lực đẩy từ các drone xung quanh (được coi là vật cản động). Điều này cho phép thuật toán phản ứng tức thời với những thay đổi cục bộ trong quá trình di chuyển, đảm bảo duy trì khoảng cách an toàn một cách tự nhiên và liên tục.
- Hiệu suất tính toán cao: Việc tính toán lực tại mỗi bước chỉ phụ thuộc vào vị trí hiện tại của drone và các drone lân cận, thay vì phải tìm kiếm toàn cục trong toàn bộ không gian. Bản chất song song tiềm năng này giúp APF đặc biệt phù hợp để xử lý bài toán với số lượng drone lớn và mở ra khả năng tăng tốc đáng kể thông qua

các nền tảng tính toán hiệu năng cao như GPU một yếu tố then chốt mà đồ án này hướng tới.

- Tạo đường đi mượt mà: Vector lực tổng hợp từ APF tự nhiên dẫn đến các quỹ đạo liên tục và ít thay đổi hướng đột ngột, phù hợp với khả năng bám quỹ đạo của drone thực tế, giúp tiết kiệm năng lượng và nâng cao tính ổn định của màn trình diễn.

Mặc dù thuật toán APF cổ điển tồn tại một số hạn chế như dễ rơi vào cực tiểu cục bộ hoặc gây dao động khi drone di chuyển gần biên của trường thê, các nghiên cứu gần đây (như của Sun [10] mà đồ án kế thừa) đã đề xuất nhiều cải tiến hiệu quả. Bằng cách kết hợp APF với các cơ chế như phân lớp trường đầy, tích hợp ràng buộc động học, và thuật toán hoán đổi mục tiêu, những nhược điểm này có thể được khắc phục một cách triệt để. Với những lý do trên, APF cùng các cải tiến được trình bày trong mục tiếp theo được đánh giá là một lựa chọn tối ưu, cung cấp một giải pháp cân bằng giữa hiệu suất, tính an toàn và khả năng triển khai thực tế cho bài toán quy hoạch đường đi chuyển đổi đối hình drone quy mô lớn. Một khác, nếu số lượng drone không quá lớn với trạm mặt đất có hiệu năng tính toán cao cũng có thể áp dụng phương pháp này theo chế độ online nhằm tạo ra các màn trình diễn có tính linh hoạt cao như trong nghiên cứu [4].

2.2.2. Thuật toán trường thê năng nhân tạo

Thuật toán trường thê năng nhân tạo là một phương pháp quy hoạch đường đi nhằm mục đích điều khiển cánh tay robot và robot di động được đề xuất bởi O.Khatib [15] Thuật toán này sử dụng một trường lực ảo để điều khiển robot trong không gian ý tưởng cơ bản của phương pháp này là:

Xác định các tọa độ của robot, mục tiêu robot hướng đến, và các vật cản $\mathbf{R}, \mathbf{G}, \mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_n$, tạo cấu trúc trường lực hút nhằm đưa robot tiến đến đích theo công thức 2.8, lực hấp dẫn tỉ lệ thuận theo khoảng cách của robot và mục tiêu theo công thức 2.7.

$$U(G) = \frac{1}{2} \varepsilon \|\mathbf{G} - \mathbf{R}\|^2 \quad (2.8)$$

$$U(O_n) = \begin{cases} \frac{1}{2} \eta \left(\frac{1}{\|\mathbf{O}_n - \mathbf{R}\|} - \frac{1}{\rho} \right)^2, & \text{nếu } \|\mathbf{O}_n - \mathbf{R}\| \leq \rho, \\ 0, & \text{nếu } \|\mathbf{O}_n - \mathbf{R}\| \geq \rho. \end{cases} \quad (2.9)$$

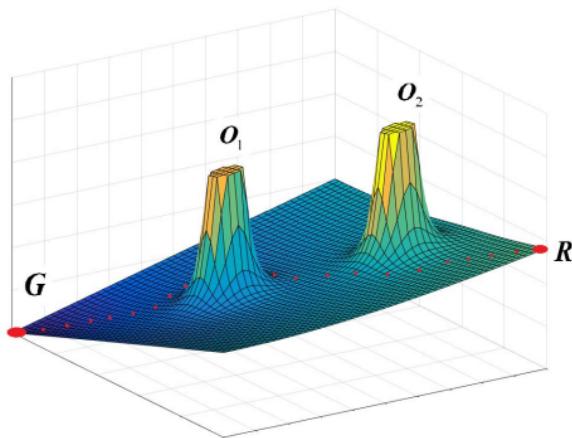
Với $U(G)$, $U(O)$ là trường hấp dẫn và trường đẩy, ε là hệ số hấp dẫn và η là hệ số trường đẩy, ρ là bán kính an toàn của robot. Nếu có n vật cản thì trường đẩy sẽ được tính bằng tổng của trường đẩy phản ứng với n vật cản suy ra được trường thê năng tổng hợp được tính như công thức 2.9

$$U(R) = U(G) + U(O_1) + U(O_2) + \dots + U(O_n). \quad (2.10)$$

Sau đó lấy gradient trường thế năng để thu được một vector lực có hướng

$$\mathbf{F}(\mathbf{R}) = -\nabla U(R). \quad (2.11)$$

Tính toán các điểm trên đường đi liên tục đến khi robot chạm được đến điểm tích thì sẽ dừng quá trình tính toán, hình 2.1 là minh họa quá trình thực hiện thuật toán trường thế năng nhân tạo với bề mặt rắn mô hình trường thế và các điểm màu đỏ là các điểm nằm trên đường đi của robot. Các hệ số như ε, η cần được lựa chọn để thuật toán có thể đạt được hiệu quả cao.



Hình 2.1. Một robot được điều hướng đến đích và tránh vật cản nhờ thuật toán trường thế năng nhân tạo.

2.2.3. Thuật toán trường thế năng nhân tạo cải tiến cho trình diễn ánh sáng sử dụng drone.

Thuật toán trường thế năng có thể áp dụng cho quá trình chuyển đổi đội hình trong các màn trình diễn ánh sáng với mô hình bài toán như mục 2.2.1. Với trường hấp dẫn và trường đẩy được tính toán trong công thức 2.11 và 2.12, với một drone bay gần m drone khác thì trường thế năng tổng và vector lực được tính toán như công thức 2.13 và 2.14 với vị trí của drone được xem là vật cản ở thời điểm t đang xét là \mathbf{p}_O^t .

$$U_{att_i} = \frac{1}{2} \varepsilon \|\mathbf{p}_i^e - \mathbf{p}_i^t\|^2 \quad (2.12)$$

$$U_{rep_{i,O}} = \begin{cases} \frac{1}{2} \eta \left(\frac{1}{\|\mathbf{p}_i^t - \mathbf{p}_O^t\|} - \frac{1}{\rho} \right)^2, & \text{nếu } \|\mathbf{p}_i^t - \mathbf{p}_O^t\| \leq \rho, \\ 0, & \text{nếu } \|\mathbf{p}_i^t - \mathbf{p}_O^t\| \geq \rho. \end{cases} \quad (2.13)$$

$$U_{res_i} = U_{att_i} + \sum_{O=1}^m \mathbf{p}_O^t \quad (2.14)$$

$$\mathbf{F}_{res_i} = -\nabla U_{res_i}; \quad (2.15)$$

Ở mỗi thời điểm t , drone xác định hướng di chuyển tiếp theo của nó dựa trên hướng của lực. Vị trí của drone tại thời điểm $t+1$ được xác định bằng cách di chuyển một khoảng cố định theo hướng đó. Giả sử λ là kích thước bước nhảy điểm tiếp theo của quãng đường sẽ được tính toán như sau.

$$\mathbf{p}_i^{t+1} = \mathbf{p}_i^t + \lambda \cdot \frac{\mathbf{F}_{res_i}^t}{\|\mathbf{F}_{res_i}^t\|} \quad (2.16)$$

Do đặc tính vật lý của drone nên gia tốc và vận tốc thường được giới hạn [5], [6], do đó trong thuật toán quy hoạch đường đi các waypoint được tính toán cũng phải phù hợp với các ràng buộc động học của drone. Kích thước bước λ được tính toán dựa trên giới hạn này theo phương trình (2.16), trong quá trình bay tốc độ càng cao thì khả năng bám quỹ đạo của drone càng giảm nên khi khoảng cách càng nhỏ thì vận tốc drone càng được giảm suốt để đảm bảo an toàn do đó các giá trị như khoảng cách hiện tại của drone với điểm mục tiêu hay khoảng cách của drone với các drone khác được áp dụng đến tính toán kích thước bước nhảy lambda , khoảng cách của drone đến điểm xuất phát cũng được xem xét để tính toán kích thước bước nhảy để đảm bảo drone không bị thay đổi vận tốc đột ngột ở khoảng thời gian bắt đầu di chuyển như công thức (2.17). T là chu kì cập nhật vị trí drone [10], [11].

$$\lambda = \begin{cases} \sqrt{2ad_{min}} \cdot T & \text{nếu } d_{min} \leq \frac{v_{max}^2}{2a} \\ v_{max} \cdot T & \text{ngược lại} \end{cases} \quad (2.17)$$

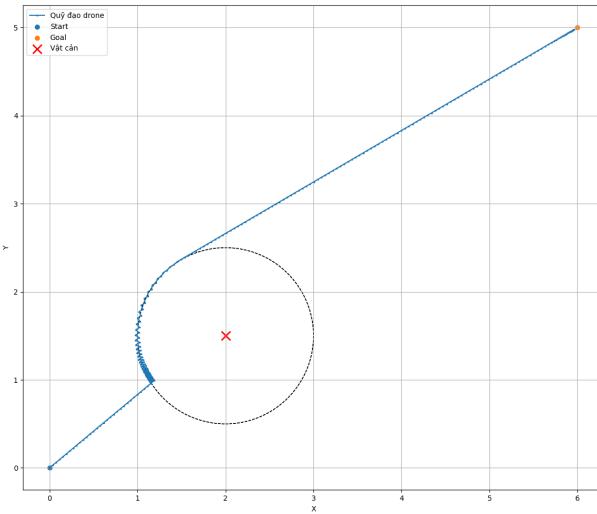
$$d_{min} = \min(\|\mathbf{p}_i^s - \mathbf{p}_i^t\|, \|\mathbf{p}_i^e - \mathbf{p}_i^t\|, \|\mathbf{p}_O - \mathbf{p}_i^t\|) \quad (2.18)$$

Trong quá trình tính toán các waypoint để đảm bảo các drone không xảy ra hiện tượng va chạm thì hệ số trường đẩy thường lớn hơn rất nhiều nếu so với hệ số hấp dẫn. Điều này dẫn đến khi drone bay vào vùng biên trường đẩy thì sẽ bị bật ra một khoảng cách xa sau đó sẽ được kéo lại về phía điểm mục tiêu điều này diễn ra liên tục sẽ gây ra cho drone những dao động không đáng có như hình 2.2. Nếu trong trường hợp trình diễn với số lượng lớn drone dày đặc thì hiện tượng này càng trở nên nghiêm trọng do việc drone này bay vào trường đẩy của drone khác sẽ diễn ra với tần suất cao hơn.

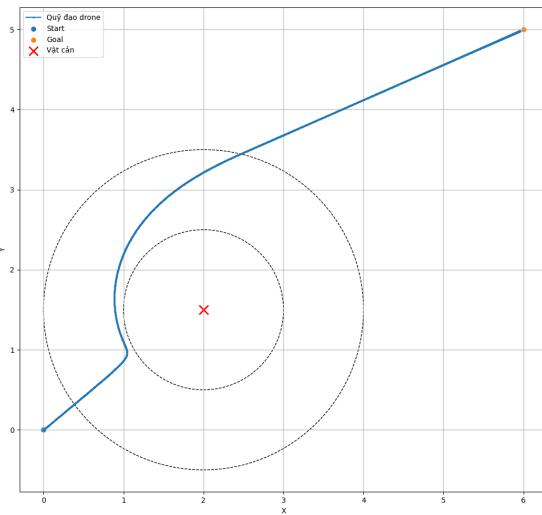
Để giảm hiện tượng này có thể cấu trúc lại lực đẩy bằng cách phân lớp như công thức 2.18. Bán kính lớp trong là ρ_1 , bán kính lớp ngoài là ρ_2 , hệ số trường đẩy ngoài là η_2 và hệ số lực hấp dẫn có cùng độ lớn với nhau, do đó khi bay vào trường đẩy ngoài, lực tác động nhỏ hướng của drone sẽ thay đổi chậm, trường đẩy ở lớp trong vẫn có hệ số η_1 lớn để đảm bảo khoảng cách an toàn cho các drone. Việc phân lớp này giúp giảm đáng kể vấn đề dao động quỹ đạo như thấy trong hình 2.3 [10]. Hình 2.2 và hình 2.3 là kết quả khi thực hiện mô phỏng với bán kính an toàn là 1 m, hệ số trường đẩy lớp trong là 100,

hệ số trường đẩy lớp ngoài là 5, bán kính lớp ngoài là 2 m, hệ số trường hấp dẫn là 2.

$$U_{\text{rep},o} = \begin{cases} \frac{1}{2}\eta_1 \left(\frac{1}{\|\mathbf{p}_o^t - \mathbf{p}_i^t\|} - \frac{1}{\rho_1} \right)^2, & \|\mathbf{p}_o^t - \mathbf{p}_i^t\| \leq \rho_1 \\ \frac{1}{2}\eta_2 \left(\frac{1}{\|\mathbf{p}_o^t - \mathbf{p}_i^t\|} - \frac{1}{\rho_2} \right)^2, & \rho_1 < \|\mathbf{p}_o^t - \mathbf{p}_i^t\| \leq \rho_2 \\ 0, & \|\mathbf{p}_o^t - \mathbf{p}_i^t\| > \rho_2 \end{cases} \quad (2.19)$$



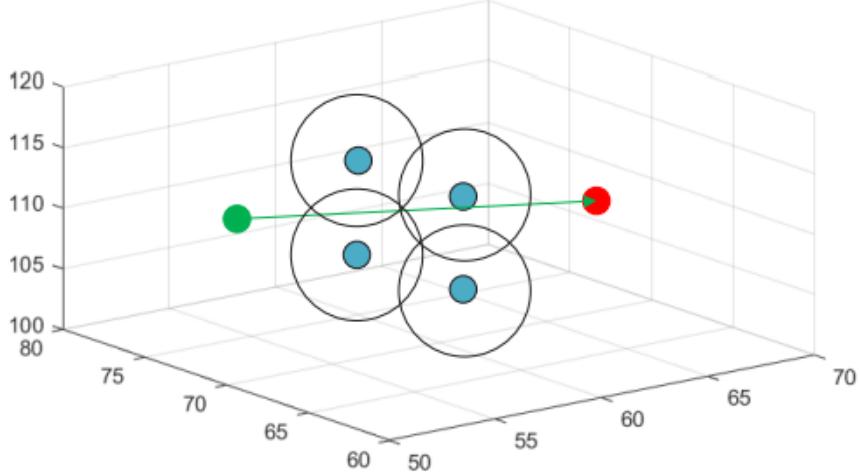
Hình 2.2. Quãng đường drone bị dao động



Hình 2.3. Quãng đường drone được phân lớp

Khi có nhiều drone khác tập trung gần điểm mục tiêu của drone đang xét đặc biệt là trong đó có nhiều drone đã đến được mục tiêu được phân công, dưới tác động của nhiều lực đẩy và lực hấp dẫn khác nhau có thể hình thành một điểm tối ưu cục bộ (local optimal solution) điều này dẫn đến trạng không thể tiến tới điểm tối ưu đã được phân công và

bị kẹt trong trường thế năng của các drone khác. Trong nghiên cứu này thì tất cả drone đều hoạt động giống nhau, nên khi một drone đang bị mắc kẹt có thể hoán đổi mục tiêu với các drone khác cũng đang bị kẹt từ đó có thể giải quyết vấn đề nghiệm tối ưu cục bộ mà không cần thay đổi đội hình drone đã được thiết kế. Các điều kiện hoán đổi được xây dựng chặt chẽ đảm bảo chỉ hoán đổi khi drone không thể tiến tới mục tiêu. Các điều kiện được thiết lập như sau [10]. Hiện tượng mắc kẹt và hoán đổi mục tiêu được thể hiện ở hình 2.4.



Hình 2.4. Minh họa hiện tượng hoán đổi vị trí

Điều kiện thứ nhất có ít nhất một drone d bước vào phạm vi của ít nhất hai drone khác đã hoàn thành nhiệm vụ d. Giả sử các drone khác là $a_1, a_2, a_3, \dots, a_n$.

$$\begin{aligned} & \exists \left(\left\| \mathbf{p}_d^t - \mathbf{p}_{a_1}^t \right\| < \rho_2 \right) \wedge \left(\mathbf{p}_{a_1}^t = \mathbf{p}_{a_i}^e \right) \\ & \quad \vdots \\ & \left(\left\| \mathbf{p}_d^t - \mathbf{p}_{a_n}^t \right\| < \rho_2 \right) \wedge \left(\mathbf{p}_{a_n}^t = \mathbf{p}_{a_n}^e \right), \quad n \geq 2 \end{aligned} \quad (2.20)$$

Điều kiện thứ hai có ít nhất một drone tạo trường đẩy giữa drone d và mục tiêu của nó. Giả sử các drone trong điều kiện này là $b_1, b_2, b_3, \dots, b_n$

$$\begin{aligned} & \exists \left(\left\| \mathbf{p}_{b_1}^t - \mathbf{p}_e^t \right\| < \left\| \mathbf{p}_d^t - \mathbf{p}_e^t \right\| \right) \\ & \quad \vdots \\ & \left(\left\| \mathbf{p}_{b_n}^t - \mathbf{p}_e^t \right\| < \left\| \mathbf{p}_d^t - \mathbf{p}_e^t \right\| \right) \quad n \geq 1 \\ & \{b_1, b_2, \dots, b_n\} \subseteq \{a_1, a_2, \dots, a_n\} \end{aligned} \quad (2.21)$$

Điều kiện thứ ba là khoảng cách giữa vị trí hiện tại của drone d đến mục tiêu lớn hơn khoảng cách giữa vị trí của drone d với mục tiêu ở vòng lặp trước đó

$$\left\| \mathbf{p}_d^{t-1} - \mathbf{p}_d^e \right\| < \left\| \mathbf{p}_d^t - \mathbf{p}_d^e \right\| \quad (2.22)$$

Nếu cả ba điều kiện này đều thỏa mãn đồng thời thì kích hoạt thuật toán hoán đổi mục tiêu, phương pháp hoán đổi là trong các drone thỏa mãn điều kiện hai chọn drone gần với drone hiện đang mắc kẹt trong nghiệm tối ưu cục bộ. Gọi drone này là r được xác định như phương trình (2.23), việc hoán đổi mục tiêu của drone r với mục tiêu của drone d giúp phá hủy trường đẩy đang cản trở làm drone d bị mắc kẹt giúp quá trình chuyển đổi được hoàn tất có thể quan sát như ở hình 2.4

$$\|\mathbf{p}_r^t - \mathbf{p}_d^t\| = \min \|\mathbf{p}_{b_i}^t - \mathbf{p}_d^t\| r, b_i \in b_1, b_2, b_3, \dots, b_n \quad (2.23)$$

2.3. Khung chương trình tính toán điểm điều hướng được đề xuất.

Các đội hình sẽ được thiết kế bởi các phần mềm chuyên dụng như blender xong xuất ra các waypoint các waypoint này sẽ làm đầu vào cho thuật toán phân công nhiệm vụ bay như đã trình bày trong thuật toán 1. Kết quả phân công tối ưu tìm được như một lớp tối ưu toàn cục, đảm bảo chi phí chuyển đổi đội hình được giảm đáng kể. Sau khi hoàn tất phân công thì thuật toán trường thế năng nhân tạo cải tiến sẽ tính toán đường đi theo chu kì. Ở mỗi bước, drone được điều khiển dựa trên lực hút tới mục tiêu và lực đẩy từ các drone xung quanh, đồng thời kích thước bước di chuyển được điều chỉnh theo các ràng buộc động học để đảm bảo an toàn và bám quỹ đạo mượt mà. Khi xảy ra hiện tượng mắc kẹt cục bộ cơ chế hoán đổi mục tiêu sẽ được kích hoạt để đảm bảo tất cả drone đều đạt được vị trí yêu cầu. Tuy nhiên có thể thấy trong công thức (2.17) và (2.18) ở λ sẽ không thể khởi tạo do đó cần khởi tạo một giá trị lambda đủ nhỏ để không thể phá hủy ràng buộc về động học của drone như công thức 2.24. Chương trình tính toán các điểm điều hướng theo chu kì được trình bày như hình 2.5, các giá trị như gia tốc, vận tốc và khoảng cách nhỏ nhất với các drone khác của drone i sẽ được tính toán ở mỗi bước lặp để đánh giá sự ổn định của phương pháp như các công thức 2.25, 2.26, 2.27. Trong quá trình tính toán để tránh trường hợp drone vượt quá xa mục tiêu thì khi một drone cách mục tiêu giá trị λ tính được thì ngay bước lặp tiếp theo vị trí của drone sẽ được gán thẳng đến mục tiêu như trên hình 2.5.

$$\lambda_k = a_{max} \cdot T^2 \quad (2.24)$$

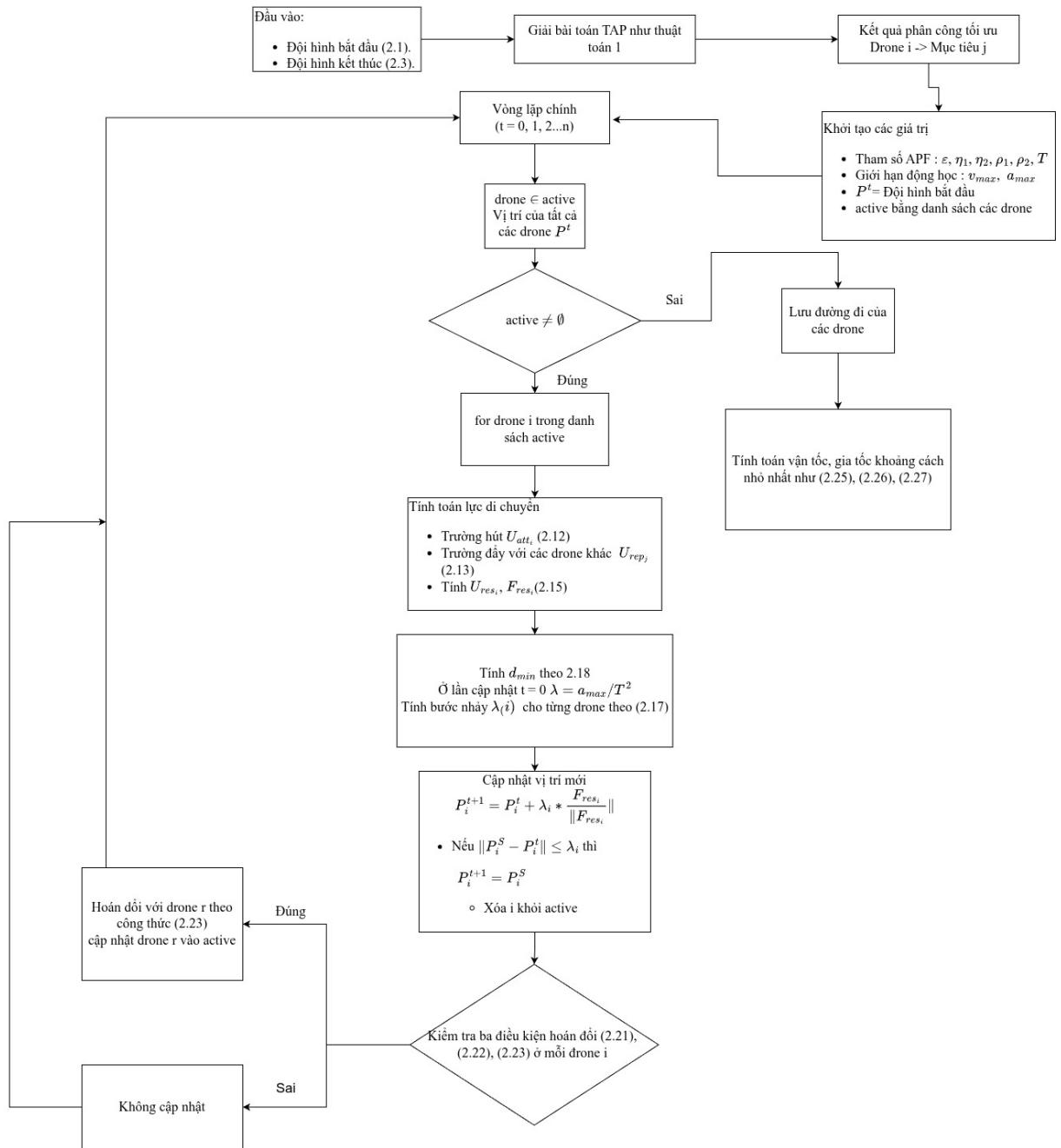
$$v_i^t = \|\mathbf{p}_i^t - \mathbf{p}_i^{t-1}\| \cdot T \quad (2.25)$$

$$a_i^t = \|a_i^t - a_i^{t-1}\| \cdot T \quad (2.26)$$

$$d = \min \left\| \mathbf{p}_i^t - \mathbf{p}_j^t \right\|, j \neq i \quad i, j \in \{1, \dots, n\} \quad (2.27)$$

Nhìn hình 2.5 có thể thấy được thời gian tính toán tăng lên đáng kể khi tăng số lượng drone tính toán tăng lớn nên khi thực hiện màn trình diễn với số lượng drone do lúc này thuật toán trường thế năng nhân tạo cải tiến sẽ phải xử lý một lượng lớn các kết

quả ở mỗi vòng lặp. Khoảng cách điểm bắt đầu với điểm mục tiêu cũng ảnh hưởng đến thời gian tính toán do phải tính toán qua nhiều bước lặp hơn nên việc phân công tối ưu cũng làm giảm thời gian tính toán cho thuật toán quy hoạch đường đi. Danh sách active trong hình 2.5 là danh sách các drone chưa hoàn thành nhiệm vụ bay trong quá trình chuyển đổi.



Hình 2.5. Khung chương trình đề xuất

Chương 3.

Lập trình song song GPU tăng tốc cho thuật toán quy hoạch đường đi.

3.1. Đặt vấn đề

Trong hệ thống trình diễn ánh sáng bằng drone, việc lập kế hoạch đường đi (path planning) là bước tiền xử lý quan trọng được thực hiện offline trước buổi diễn. Như đã trình bày ở Chương 2, thuật toán APF cải tiến có độ phức tạp tính toán $O(n^2)$ cho mỗi bước lặp khi triển khai tuần tự trên CPU, với n là số lượng drone. Khi mở rộng quy mô lên hàng trăm đến hàng nghìn drone, thời gian tính toán cần thiết để tạo ra toàn bộ quy đạo có thể trở nên quá lớn, gây khó khăn cho quá trình thiết kế, tối ưu hóa và mô phỏng các màn trình diễn phức tạp. Chẳng hạn, với một kịch bản chuyển đổi đội hình có 1.000 drone, mỗi bước lặp đòi hỏi khoảng 10^6 phép tính. Nếu quá trình chuyên đổi cần 1.000 bước lặp, tổng số phép tính lên tới 10^9 . Trên CPU thông thường, thời gian xử lý có thể kéo dài hàng giờ, làm chậm trễ đáng kể quy trình chuẩn bị và hạn chế khả năng thử nghiệm nhiều phương án thiết kế đội hình khác nhau. Mặc dù đây là bài toán offline, việc giảm thời gian tiền xử lý vẫn có ý nghĩa thực tiễn cao, cho phép

- Tăng tốc quy trình thiết kế: Người thiết kế có thể nhanh chóng mô phỏng và điều chỉnh các kịch bản phức tạp.
- Mở rộng quy mô mô phỏng: Khả năng tính toán cho số lượng drone lớn hơn, tạo ra những màn trình diễn hoành tráng hơn.
- Tối ưu hóa tham số: Dễ dàng thực hiện nhiều lần chạy mô phỏng với các bộ tham số khác nhau để tìm phương án tối ưu.

Lập trình song song trên GPU (Graphics Processing Unit) xuất hiện như một giải pháp khả thi nhờ kiến trúc bao gồm hàng nghìn lõi xử lý, tối ưu cho các tác vụ thực thi cùng một thao tác trên nhiều luồng dữ liệu độc lập (mô hình SIMD) [16]. Bản chất của thuật toán APF với các phép tính lực cho từng drone có thể được thực hiện độc lập và đồng thời rất phù hợp để song song hóa trên GPU. Chương này trình bày giải pháp tăng tốc thuật toán APF cải tiến thông qua lập trình song song trên GPU sử dụng PyOpenCL, nhằm giảm đáng kể thời gian tính toán offline, từ đó hỗ trợ hiệu quả cho việc lập kế hoạch và mô phỏng các màn trình diễn ánh sáng drone quy mô lớn.

3.2. Giới thiệu về OpenCL và PyOpenCL

3.2.1. Lý do lựa chọn PyOpenCL cho đồ án.

OpenCL (Open Computing Language) là một framework lập trình song song đa nền tảng, cho phép khai thác sức mạnh tính toán của các thiết bị như GPU, CPU và FPGA thông qua một mô hình lập trình thống nhất [17]. Kiến trúc của OpenCL dựa trên việc thực thi một hàm (kernel) bởi nhiều luồng xử lý song song (work-items) được tổ chức trong một không gian chỉ số đa chiều (NDRange). Các work-item có thể được nhóm thành work-group để chia sẻ bộ nhớ cục bộ (local memory) và đồng bộ với nhau. Trong bối cảnh đồ án này việc lựa chọn OpenCL thông qua thư viện PyOpenCL dựa nhiều vào tính đa nền tảng có thể triển khai trên nhiều loại phần cứng khác nhau. Đặc biệt, đối với card đồ họa AMD Radeon RX thuộc dòng gaming được sử dụng trong đồ án, OpenCL là framework tính toán song song khả thi nhất do Cuda là độc quyền của Nvidia [16] không hỗ trợ phần cứng AMD, trong khi đó ROCm (nền tảng tính toán mở của AMD) có hỗ trợ hạn chế trên card gaming.

PyOpenCL là một thư viện Python cung cấp giao diện lập trình cho OpenCL, cho phép các nhà phát triển viết mã OpenCL trong môi trường Python dễ sử dụng và linh hoạt [18]. PyOpenCL hỗ trợ đầy đủ các tính năng của OpenCL, bao gồm quản lý bộ nhớ, biên dịch kernel, và thực thi kernel trên các thiết bị tính toán cũng như dễ dàng kết hợp với các thư viện khoa học khác trong hệ sinh thái Python như NumPy, SciPy để xử lý dữ liệu hiệu quả, PyOpenCL giúp đơn giản hóa quá trình phát triển ứng dụng song song, giảm bớt độ phức tạp khi làm việc trực tiếp với API OpenCL cấp thấp. Đối với đồ án tốt nghiệp này, việc chứng minh được nguyên lý tăng tốc bằng GPU quan trọng hơn việc đạt hiệu suất tuyệt đối. PyOpenCL cung cấp một nền tảng đủ tốt để minh họa nguyên lý này, đồng thời cho phép mã nguồn dễ dàng được kiểm chứng và chạy trên nhiều hệ thống khác nhau.

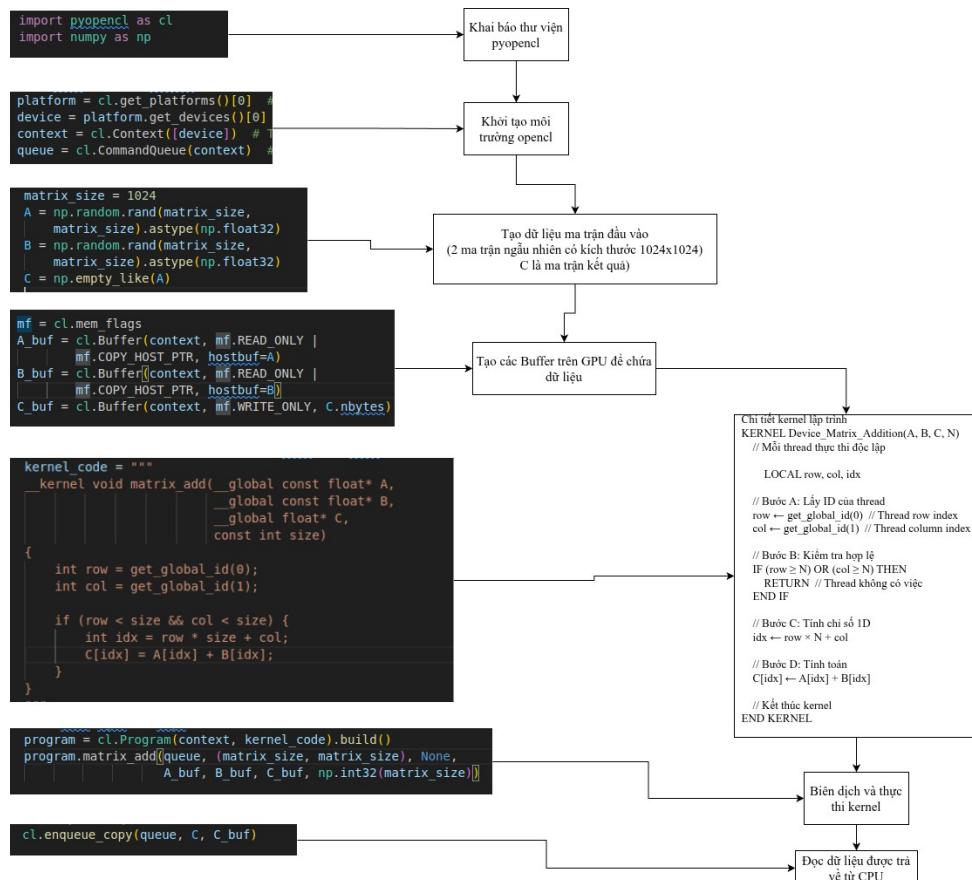
3.2.2. Kiến trúc lập trình OpenCL.

Kiến trúc lập trình PyOpenCL bao gồm các thành phần chính sau đây:

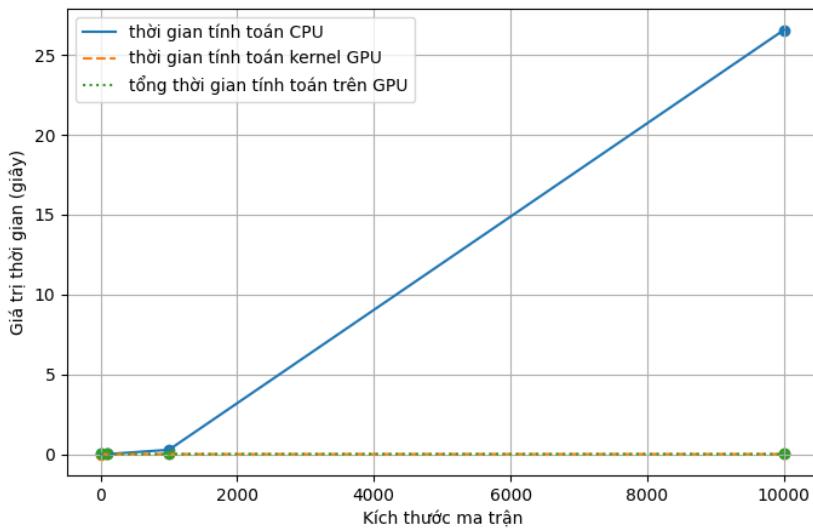
- Platform: Đại diện cho một nhà cung cấp phần cứng/phần mềm OpenCL (ví dụ: AMD, NVIDIA, Intel).
- Device: Thiết bị tính toán cụ thể (GPU, CPU) thuộc một platform nhất định.
- Context: Môi trường thực thi chứa các thiết bị và quản lý tài nguyên như thiết bị, bộ nhớ, chương trình.
- Command Queue: Hàng đợi lệnh để điều phối thực thi kernel và thao tác bộ nhớ.
- Program: Chứa mã nguồn kernel OpenCL được biên dịch để chạy trên thiết bị.
- Kernel: Hàm được thực thi song song trên nhiều work-item.

- Buffer: : Đối tượng bộ nhớ để chứa dữ liệu giữa host (CPU) và device (GPU).

Để minh họa quy trình lập trình với PyOpenCL có thể thực hiện một ví dụ đơn giản là cộng ma trận, xét bài toán cộng hai ma trận kích thước $N \times N$. Mỗi phần tử của ma trận kết quả được tính độc lập, tạo điều kiện lý tưởng cho song song hóa. Hình 3.1 minh họa quy trình lập trình với PyOpenCL giải bài toán cộng hai ma trận. Đầu tiên, ta khởi tạo OpenCL platform và device, tạo context và command queue để quản lý tài nguyên và điều phối lệnh. Tiếp theo, ta tạo buffer trên thiết bị để lưu trữ dữ liệu ma trận A, B và kết quả C. Sau đó, ta viết mã kernel OpenCL để thực hiện phép cộng ma trận. Mỗi work-item sẽ tính toán một phần tử của ma trận kết quả dựa trên chỉ số toàn cục (global ID). Tiếp theo, ta biên dịch chương trình và tạo kernel từ mã nguồn. Cuối cùng, ta thiết lập các đối số kernel, xác định kích thước NDRange và thực thi kernel trên thiết bị. Sau khi hoàn tất, ta đọc dữ liệu kết quả từ buffer trên thiết bị về host để sử dụng. Tăng tốc tính toán được thể hiện rõ ràng khi so sánh thời gian thực thi trên GPU với việc thực hiện tuần tự trên CPU, đặc biệt khi kích thước ma trận N lớn như hình 3.2.



Hình 3.1. Quy trình lập trình với PyOpenCL



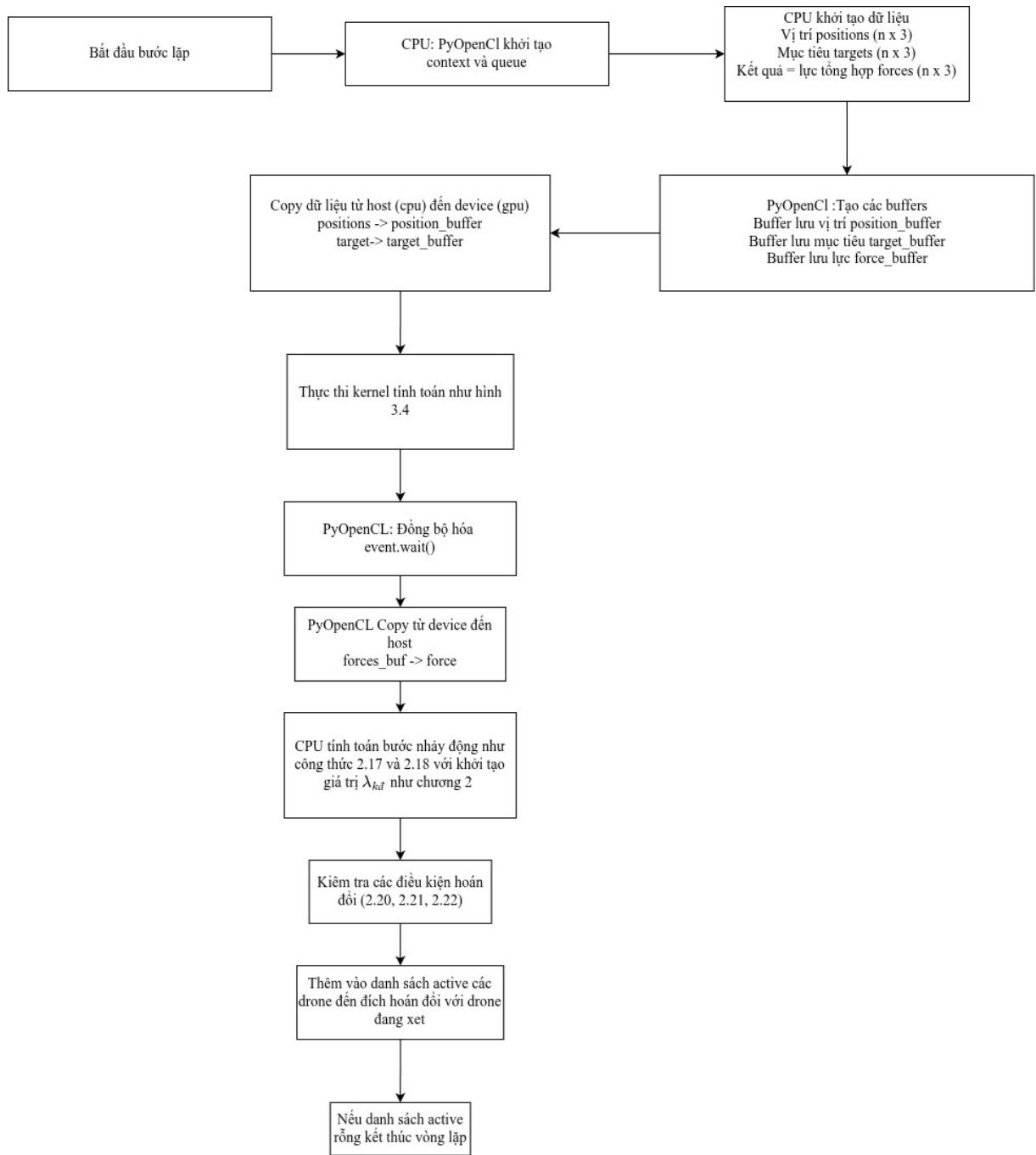
Hình 3.2. So sánh thời gian thực thi cộng ma trận trên CPU và GPU

3.3. Lập trình song song GPU cho khung chương trình để xuất.

3.3.1. Lựa chọn thành phần song song hóa.

Thuật toán trường thế năng nhân tạo cải tiến cho quy hoạch đường đi chuyển đổi đội hình drone như đã trình bày ở chương 2 được cấu thành từ nhiều thành phần có thể song song hóa. Qua quá trình phân tích, đồ án này chỉ song song hóa thành phần tính toán lực tổng hợp trên GPU ở mỗi bước lặp, các thành phần khác như tính bước nhảy λ , cập nhật vị trí của các drone hay kiểm tra điều kiện hoán đổi sẽ được thực hiện trên CPU. Quyết định này cân bằng giữa hiệu năng, độ phức tạp khi lập trình và thời gian phát triển đồ án. Về mặt lý thuyết thời gian tính toán lực tổng hợp trên CPU có độ phức tạp $O(n^2)$ chiếm phần lớn thời gian thực thi của thuật toán quy hoạch đường đi. Đây là phần tính toán nặng nhất và có đặc điểm song song hóa lý tưởng: mỗi drone đều thực hiện cùng một loại phép tính độc lập với nhau, chỉ dựa trên dữ liệu đọc chung là vị trí hiện tại của tất cả drone. Kiến trúc SIMD của GPU, với khả năng thực thi hàng nghìn thread song song, đặc biệt phù hợp để xử lý phần tính toán này một cách hiệu quả. Ngược lại các thành phần như tính toán bước nhảy hay cập nhật vị trí có độ phức tạp thấp hơn chỉ $O(n)$ do đó việc đưa chúng lên GPU không mang lại cải thiện hiệu năng đáng kể so với phần tính toán lực tổng hợp. Hơn nữa, thành phần kiểm tra và hoán đổi mục tiêu lại có logic phức tạp, yêu cầu truy cập và đánh giá toàn cục giữa các drone, khiến nó khó song song hóa hiệu quả trên GPU do gây ra hiện tượng phân kỳ nhánh (branch divergence) nghiêm trọng và đòi hỏi các thao tác atomic phức tạp.

Xét về góc độ thực tế triển khai, việc giữ các phần tính toán còn lại trên CPU mang lại hai lợi ích chính: dễ dàng gỡ lỗi và điều chỉnh thuật toán, đồng thời không làm chậm chương trình vì lượng dữ liệu cần truyền giữa CPU và GPU rất nhỏ (chỉ khoảng 24KB cho 1000 drone mỗi bước tính). Hình 3.3 minh họa khung chương trình tính toán điểm điều hướng với tăng tốc GPU thông qua PyOpenCL.



Hình 3.3. Khung chương trình tính toán điểm điều hướng với tăng tốc GPU

3.3.2. Triển khai thành phần tính toán lực tổng hợp trên GPU.

Trên GPU mặc dù có thể triển khai song song hàng nghìn luồng tính toán nhưng việc truy nhặt bộ nhớ toàn cục để đọc dữ liệu vị trí của tất cả drone sẽ khiến hiệu năng chương trình giảm. Để tối ưu hóa việc truy cập bộ nhớ, đồ án áp dụng kỹ thuật chia ô (tilling) nhằm tận dụng bộ nhớ cục bộ của GPU, giúp giảm đáng kể số lần truy cập bộ nhớ toàn cục [19]. Thay vì mỗi luồng đọc trực tiếp vị trí của tất cả drone từ bộ nhớ toàn cục, dữ liệu được chia thành các khối có kích thước cố định ví dụ 16 hoặc 32 drone. Các luồng trong một nhóm sẽ tải dữ liệu của một ô vào bộ nhớ cục bộ, sau đó mỗi luồng tính toán lực tương tác giữa drone của nó với các drone trong ô đó bằng cách đọc từ bộ nhớ cục bộ vốn có tốc độ truy cập nhanh hơn nhiều so với bộ nhớ toàn cục [19]. Mục đích của kỹ thuật này là giảm thiểu số lần truy cập bộ nhớ cục bộ, tận dụng băng thông cao của bộ nhớ cục bộ từ đó tăng hiệu suất tính toán lực tổng hợp trên GPU.

Chương 4.

Thực hiện mô phỏng và phân tích kết quả

4.1. Kết quả về thời gian tính toán.

Với mô phỏng thực hiện trên máy tính cá nhân với cấu hình CPU Intel Core i7-10750H 2.60GHz, RAM 16GB và GPU AMD Radeon RX 5600M 6GB, hệ điều hành Unbuntu, sử dụng PyOpenCL để tăng tốc thuật toán quy hoạch đường đi dựa trên trường thế năng nhân tạo cải tiến. thực hiện mô phỏng chuyển đổi đội hình với số lượng drone khác nhau gồm 5, 10, 25, 125, 250 và 600 drone. Kết quả tính toán trên CPU và GPU được trình bày trong bảng 4.1 đã cho thấy được hiệu quả của việc tính toán lực tổng hợp trên GPU so với việc thực hiện tuần tự trên CPU đặc biệt với số lượng drone lớn.

Bảng 4.1. So sánh thời gian tính toán trên CPU và GPU

Số lượng drone	Thời gian tính toán trên CPU (s)	Thời gian tính toán có sử dụng GPU (s)	Thời gian tính toán lực trên CPU (s)	Thời gian tính toán kernel lực tổng hợp trên GPU (s)
5	0.04	0.06	0.01	0.03
10	0.11	0.10	0.04	0.03
25	0.48	0.31	0.23	0.05
125	10.87	5.28	5.87	0.28
250	44.2	21.19	23.81	0.3
600	446.07	178.19	266.04	0.47

Có thể thấy trên bảng 4.1 với số lượng drone nhỏ như với 5 drone thì thời gian tính toán trên CPU nhỉnh hơn GPU do GPU chưa khai thác hết thế mạnh xử lý song song của mình. GPU được thiết kế để xử lý hàng nghìn luồng (thread) cùng lúc, nhưng với ít drone, số luồng được tạo ra quá ít, không đủ để lấp đầy các lõi tính toán và che giấu độ trễ truy cập bộ nhớ. Trong khi đó, CPU với ít lõi nhưng mạnh mẽ và không có chi phí khởi tạo lớn lại xử lý hiệu quả hơn. Chỉ khi số lượng drone đủ lớn GPU mới bắt đầu phát huy ưu thế vượt trội nhờ khả năng xử lý song song hàng loạt. Khi số lượng drone tăng lên 10, 25, 125, 250 hay 600 thì thời gian tính toán trên GPU giảm đáng kể so với CPU vì lúc này thời gian tính toán lực trên CPU sẽ rất lớn trong khi thời gian truyền dữ liệu từ CPU sang GPU và ngược lại vẫn không đổi quá nhiều. Các kết quả tính toán trên CPU cũng chỉ ra thời gian tính toán của thuật toán chủ yếu tập trung vào phần tính toán lực

tổng hợp với việc luôn chiếm đến hơn nửa quá trình tính toán phù hợp với phân tích ở mục 3.3 do đó việc tăng tốc phần này trên GPU đã mang lại hiệu quả rõ rệt. Thời gian tính toán lực tổng hợp trên kernel GPU luôn duy trì ở mức rất thấp dưới 0.5s ngay cả với 600 drone, trong khi thời gian tính toán lực trên CPU tăng lên đáng kể khi số lượng drone tăng. Các mô phỏng này được thực hiện nhằm đánh giá hiệu quả của việc tăng tốc thuật toán quy hoạch đường đi bằng GPU chưa xét đến yếu tố an toàn của quá trình chuyển đổi đội hình nên những tham số của thuật toán được điều chỉnh cho nhanh đạt đích nhất có thể. Các thông số an toàn của buổi trình diễn ánh sáng bằng drone sẽ được khảo sát trong mục tiếp theo.

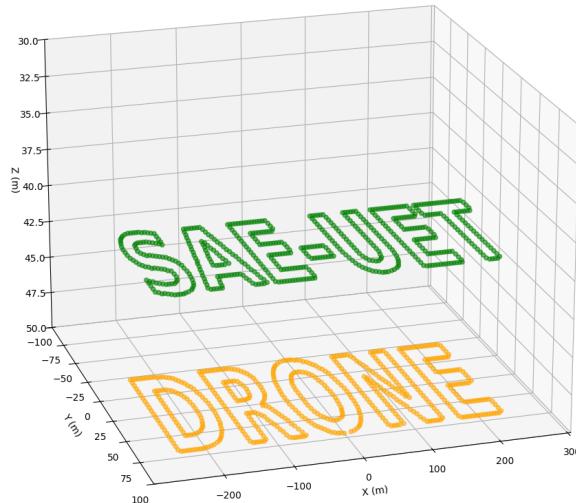
4.2. Kết quả chất lượng màn trình diễn.

4.2.1. Các thông số thuật toán ảnh hưởng đến chất lượng màn trình diễn.

Qua khung chương trình được đề xuất ở hình 2.5, các tham số của thuật toán APF như hệ số trường hấp dẫn ε , hệ số trường đẩy η_1, η_2 , bán kính lốp ngoài ρ_2 sẽ ảnh hưởng trực tiếp đến chất lượng của quá trình chuyển đổi đội hình. Trong nhiều nghiên cứu các thông số APF có thể được chọn thông qua các thuật toán metaheuristic như trong nghiên cứu của Rosas và cộng sự [1] giải thuật di truyền được sử dụng để tìm tối ưu các thông số của thuật toán APF với hàm mục tiêu là giảm tối thiểu tổng khoảng cách Euclidean từ waypoint này đến waypoint tiếp theo trên đường đi hay trong nghiên cứu [2], Szczepanski đã giải quyết bài toán tối ưu hóa đa mục tiêu để tìm tham số APF với hàm mục tiêu được thiết kế để tối thiểu thời gian đến đích, tối đa độ mượt của hướng di chuyển, giảm lực giảm tốc ở giai đoạn về đích. Tuy nhiên các nghiên cứu này thường thực hiện với số lượng robot rất hạn chế nếu áp dụng cho các đội hình trình diễn ánh sáng bằng drone với số lượng lớn thì việc áp dụng các thuật toán này khó khăn hơn rất nhiều do cần có hiệu suất tính toán cao, nên trong nghiên cứu này các tham số của thuật toán APF được lựa chọn thông qua việc khảo sát thực nghiệm với các giá trị khác nhau để đánh giá ảnh hưởng của từng tham số đến chất lượng của quá trình chuyển đổi đội hình thực chất phương pháp này cũng đã được sử dụng trong các nghiên cứu xây dựng thuật toán chuyển đổi đội hình dựa trên phương pháp trường thế năng nhân tạo như trong nghiên cứu của Sun [10] hay nghiên cứu của Huang [4], nhược điểm của phương pháp này là sẽ không đảm bảo tìm được bộ tham số tối ưu cho thuật toán nhưng phải đảm bảo được an toàn trong quá trình chuyển đổi như không xảy ra chạm và các ràng buộc động học được thỏa mãn, do đó việc khảo sát thực nghiệm là cần thiết để lựa chọn bộ tham số phù hợp cho thuật toán APF cải tiến sử dụng trong đồ án này.

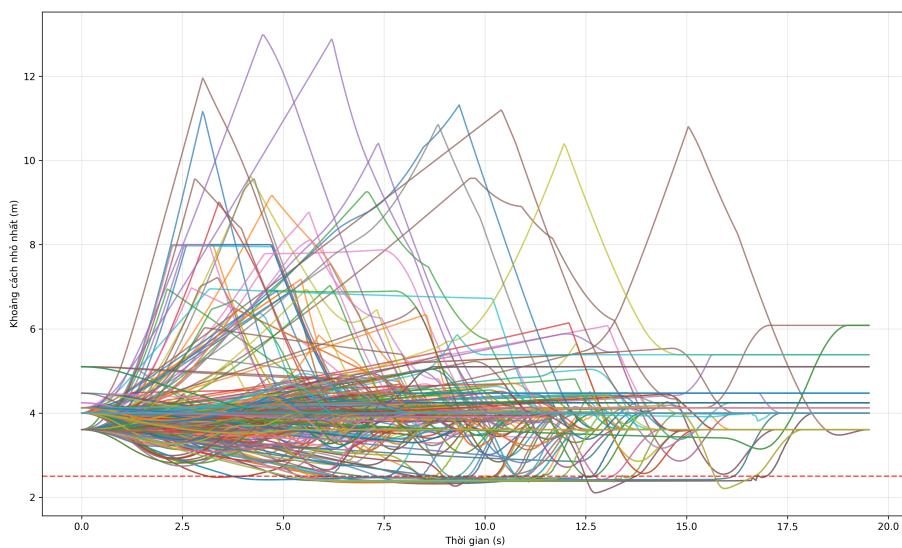
Để khảo sát, một số tham số then chốt được lựa chọn, bao gồm: hệ số trường hấp dẫn ε , hệ số trường đẩy lốp trong η_1 , hệ số trường đẩy lốp ngoài η_2 và bán kính lốp ngoài ρ_2 . Các tham số khác được cố định dựa trên đặc tính vật lý và yêu cầu an toàn

của drone: bán kính an toàn $\rho_1 = 2.5$ m, gia tốc tối đa $a_{max} = 3.0, m/s^2$, vận tốc tối đa $v_{max} = 3.0, m/s$, chu kỳ cập nhật $T = 0.04s$. Quá trình khảo sát được thực hiện trên kịch bản chuyển đổi đội hình từ chữ "SAE-UET" sang "DRONE" với 600 drone (Hình 4.1). Các chỉ tiêu đánh giá bao gồm: khoảng cách an toàn tối thiểu giữa các drone, số lần xảy ra hoán đổi mục tiêu và gia tốc trung bình của toàn đội hình.

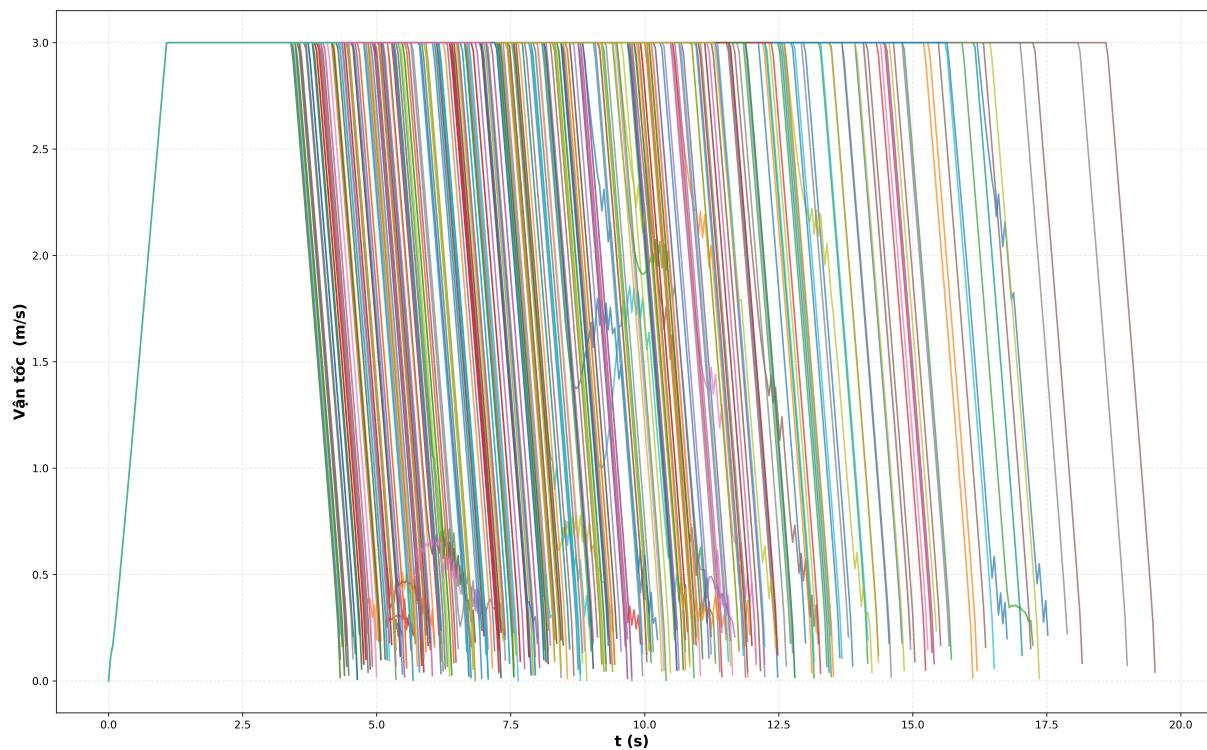


Hình 4.1. Kịch bản khảo sát

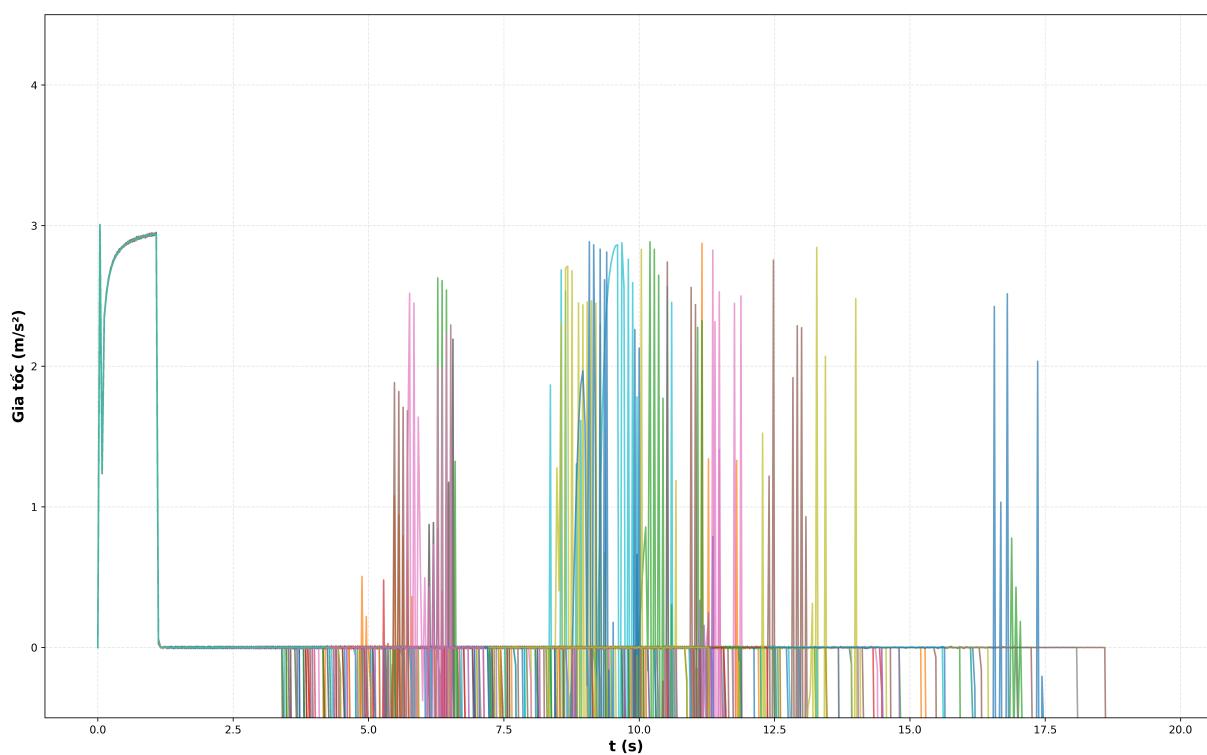
Trường hợp 1: Có thể lấy giá trị hệ số hấp dẫn tương đối lớn giúp các drone về đích nhanh $\varepsilon = 5$, hệ số trường đẩy lớp ngoài $\eta_2 = 5$, hệ số trường đẩy lớp trong rất lớn nhằm đảm bảo không có drone nào bị va chạm với nhau $\eta_1 = 500$ bán kính lớp ngoài $\rho_2 = 3.5$ m. Kết quả khảo sát các giá trị như vận tốc, gia tốc, khoảng cách nhỏ nhất giữa drone trong quá trình bay thể hiện ở hình 4.2, hình 4.3 và hình 4.4 như sau:



Hình 4.2. Khoảng cách nhỏ nhất của các drone trong quá trình chuyển đổi trong trường hợp 1

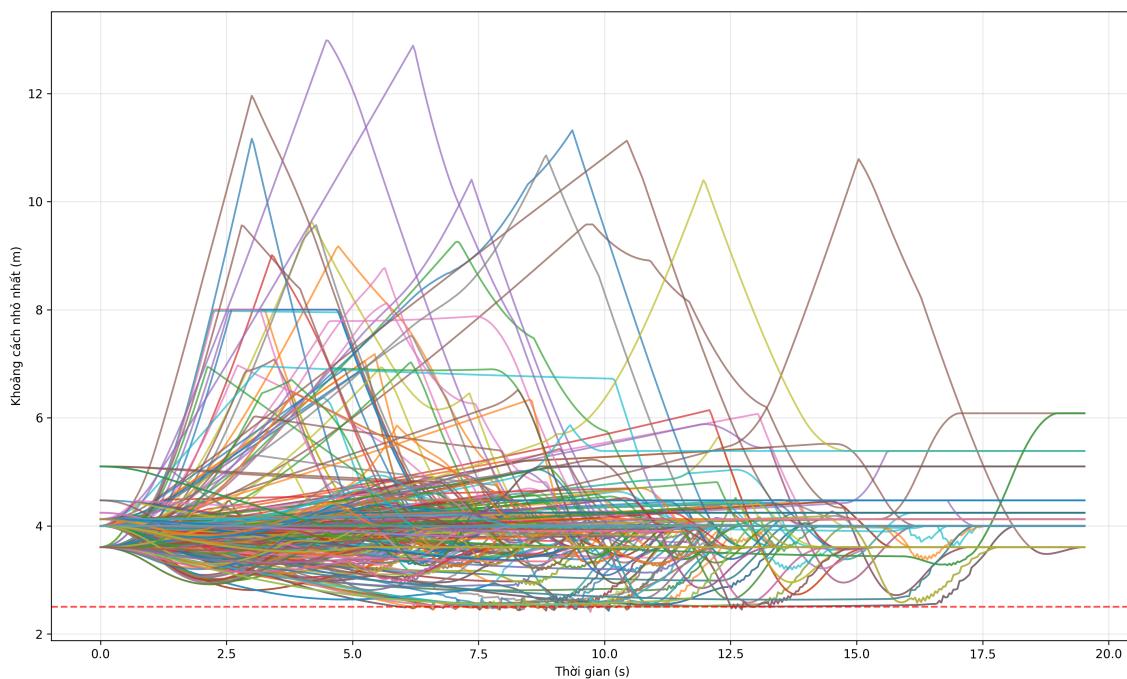


Hình 4.3. Vận tốc của các drone trong quá trình chuyển đổi trong trường hợp 1

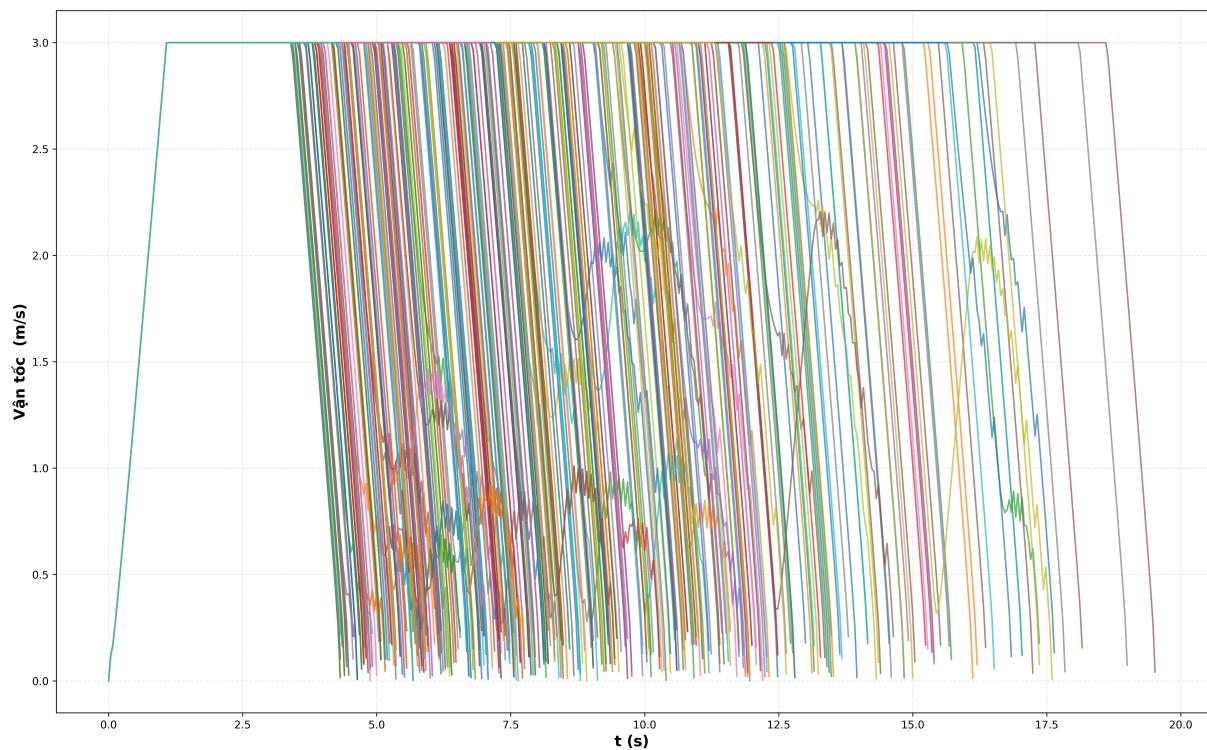


Hình 4.4. Gia tốc của các drone trong quá trình chuyển đổi trong trường hợp 1

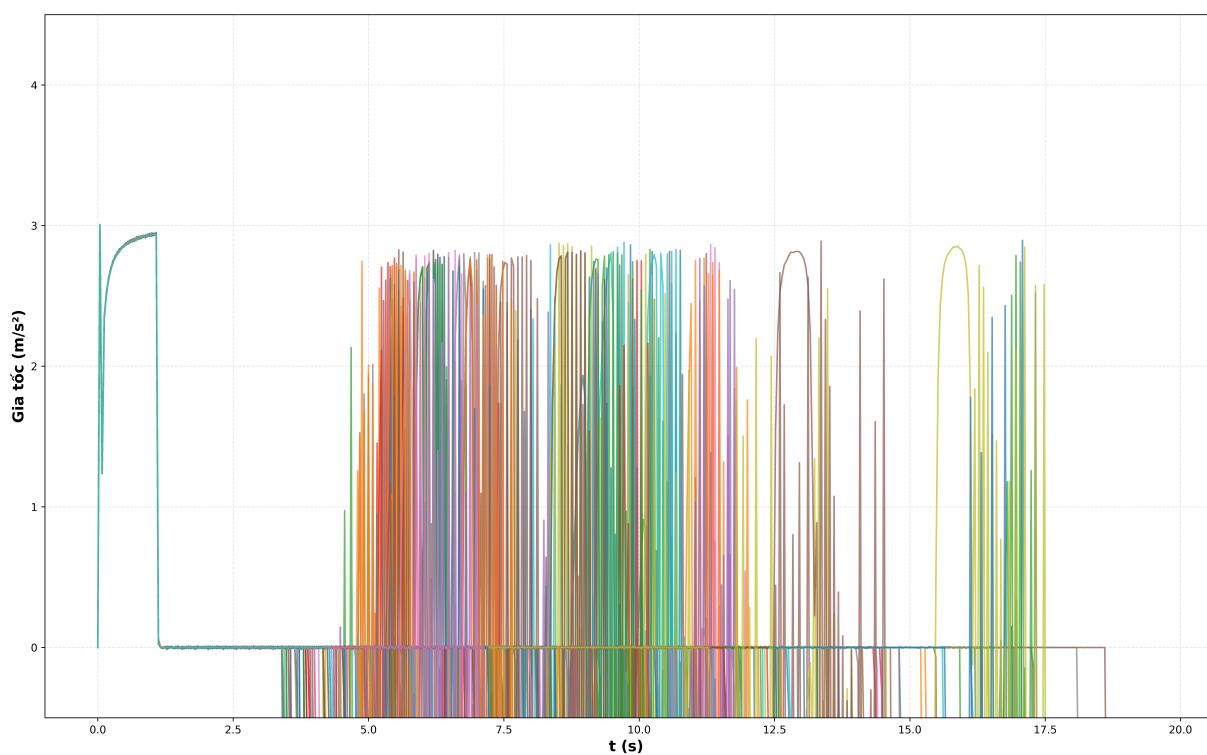
Từ các kết quả khảo sát trong trường hợp 1 có thể thấy được với việc lựa chọn hệ số trường hấp dẫn lớn giúp các drone nhanh chóng về đích tuy nhiên có thể thấy được khoảng cách giữa các drone có nhiều trường hợp bay vào bán kính an toàn của nhau vận tốc của hoạt động ở mức cao gần vận tốc tối đa điều này cho thấy được trường thê hấp dẫn đang chiếm ưu thế quá lớn so với trường thê đẩy nên có thể giảm các tham số hệ số trường hấp dẫn và tăng hệ số trường đẩy để đảm bảo an toàn hơn trong quá trình chuyển đổi đội hình. Trường hợp 2: Giảm hệ số trường hấp dẫn xuống còn $\varepsilon = 3$, tăng hệ số trường đẩy lớp ngoài lên $\eta_2 = 8$ và giữ nguyên hệ số trường đẩy lớp trong $\eta_1 = 500$ bán kính lớp ngoài vẫn giữ nguyên $\rho_2 = 3.5$ m. Kết quả khảo sát các giá trị như vận tốc, gia tốc, khoảng cách nhỏ nhất giữa drone trong quá trình bay được thể hiện ở hình 4.5, hình 4.6 và hình 4.7 như sau:



Hình 4.5. Khoảng cách nhỏ nhất của các drone trong quá trình chuyển đổi trong trường hợp 2

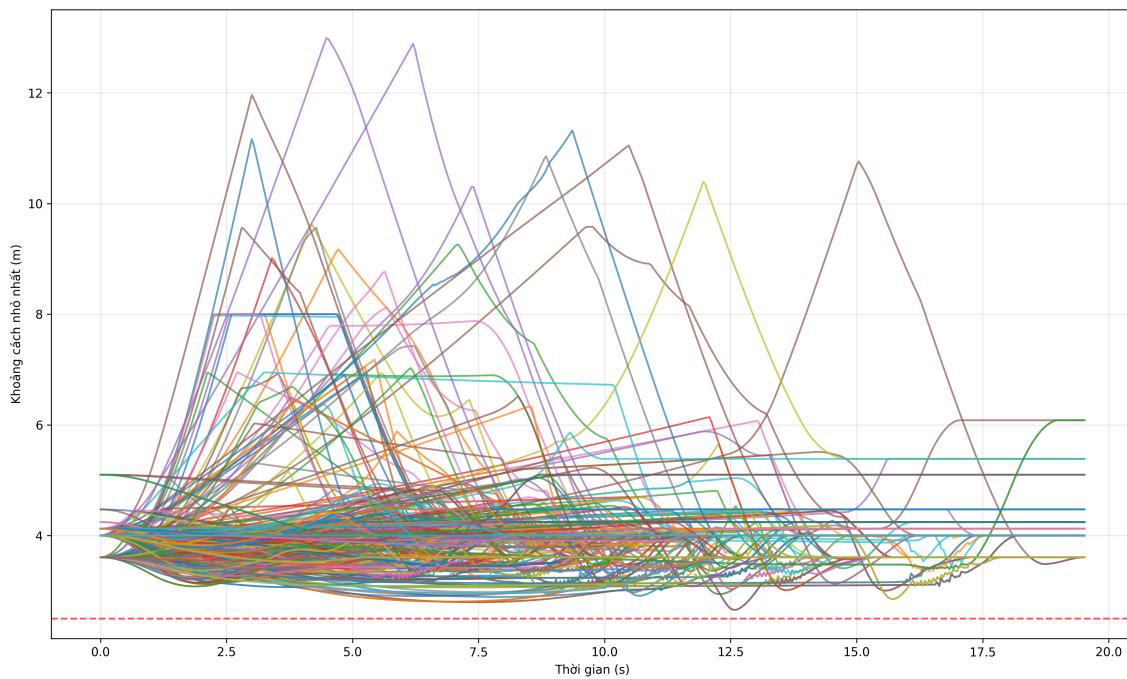


Hình 4.6. Vận tốc của các drone trong quá trình chuyển đổi trong trường hợp 2

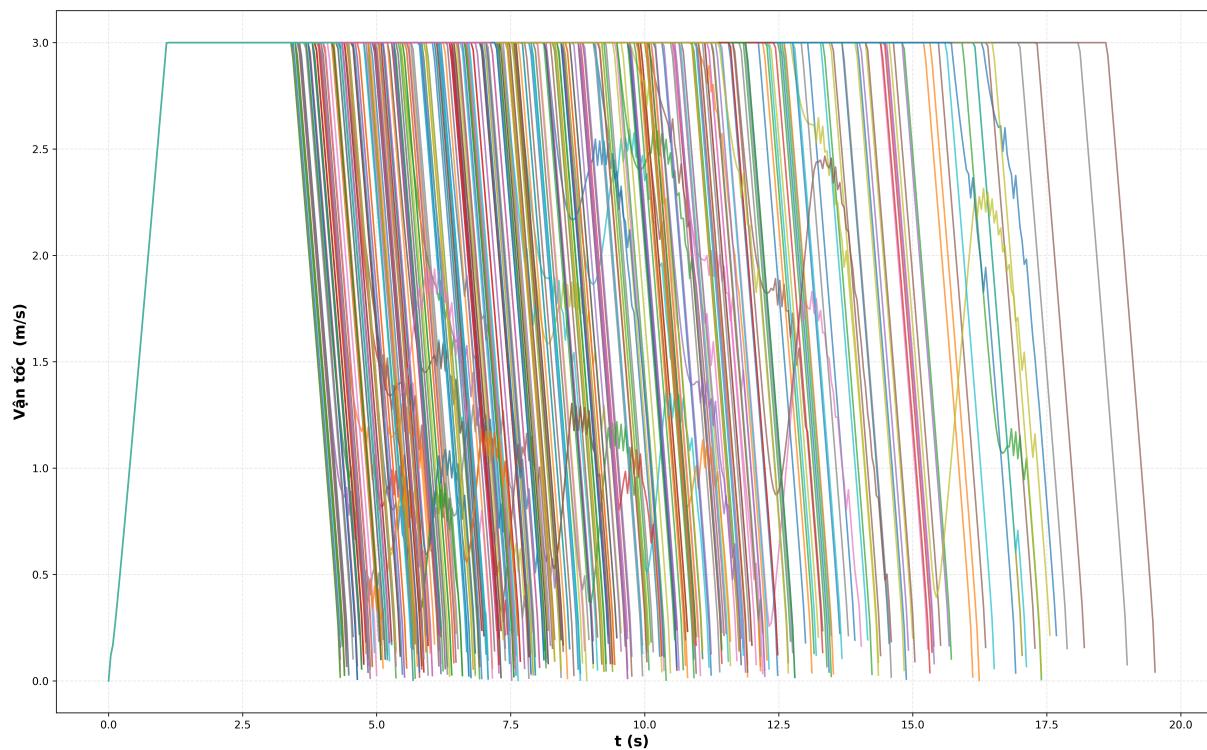


Hình 4.7. Gia tốc của các drone trong quá trình chuyển đổi trong trường hợp 2

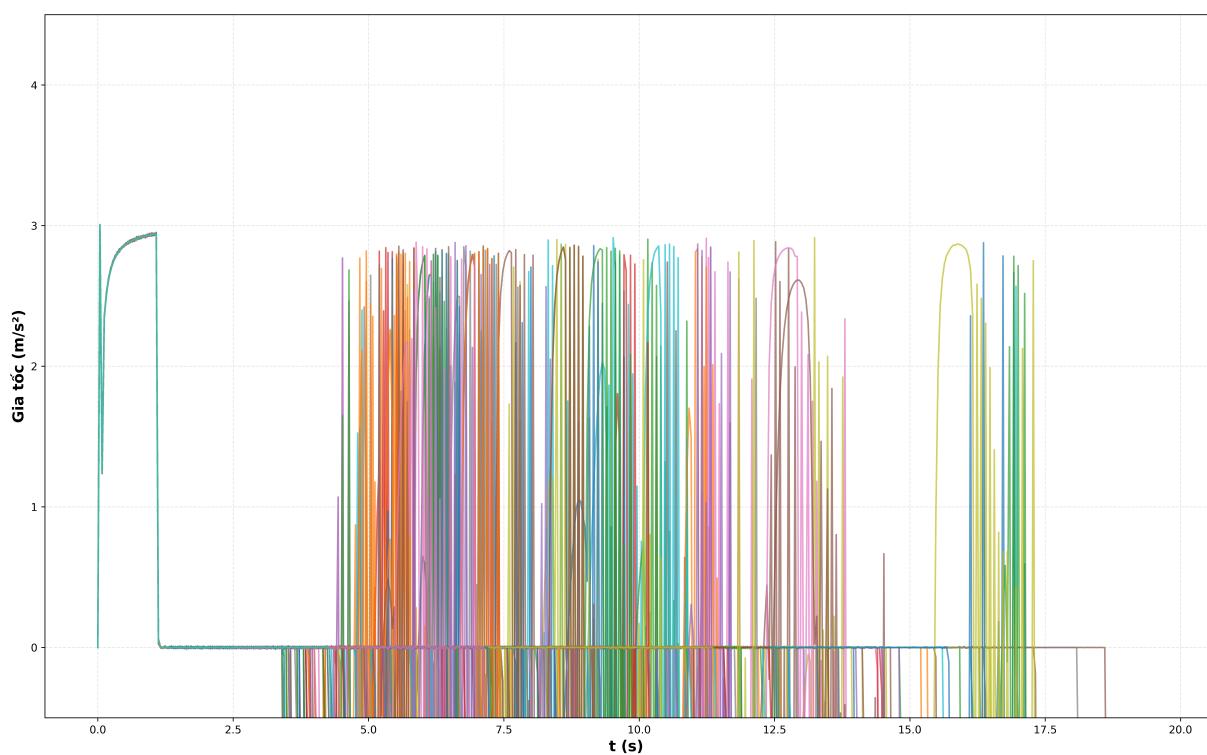
Từ các kết quả khảo sát trong trường hợp 2 có thể thấy được với việc giảm hệ số trường hấp dẫn và tăng hệ số trường đẩy giúp các drone có khoảng cách an toàn lớn hơn trong quá trình chuyển đổi đội hình nhưng vẫn còn một vài drone đi vào vùng an toàn của nhau mặc dù hệ số trường đẩy lốp trong đã được đặt rất lớn có thể làm các drone này đi xa khỏi vùng không an toàn ngay lập tức nếu tăng hệ số trường đẩy lốp ngoài lên có thể hạn chế được hiện tượng này. Trường hợp 3: Giữ nguyên hệ số trường hấp dẫn $\varepsilon = 3$, tăng hệ số trường đẩy lốp ngoài lên $\eta_2 = 30$ và giữ nguyên hệ số trường đẩy lốp trong $\eta_1 = 500$ bán kính lốp ngoài vẫn giữ nguyên $\rho_2 = 3.5$ m. Kết quả khảo sát các giá trị như khoảng cách nhỏ nhất, vận tốc và gia tốc giữa drone trong quá trình bay được thể hiện ở hình 4.8, hình 4.9 và hình 4.10. Từ các kết quả khảo sát trong trường hợp 3 có thể thấy được các drone đảm bảo được vận tốc, gia tốc cũng như không xảy ra va chạm trong quá trình chuyển đổi đội hình.



Hình 4.8. Khoảng cách nhỏ nhất của các drone trong quá trình chuyển đổi trong trường hợp 3



Hình 4.9. Vận tốc của các drone trong quá trình chuyển đổi trong trường hợp 3



Hình 4.10. Gia tốc của các drone trong quá trình chuyển đổi trong trường hợp 3

4.2.2. So sánh với kết quả của phương pháp Sun [10].

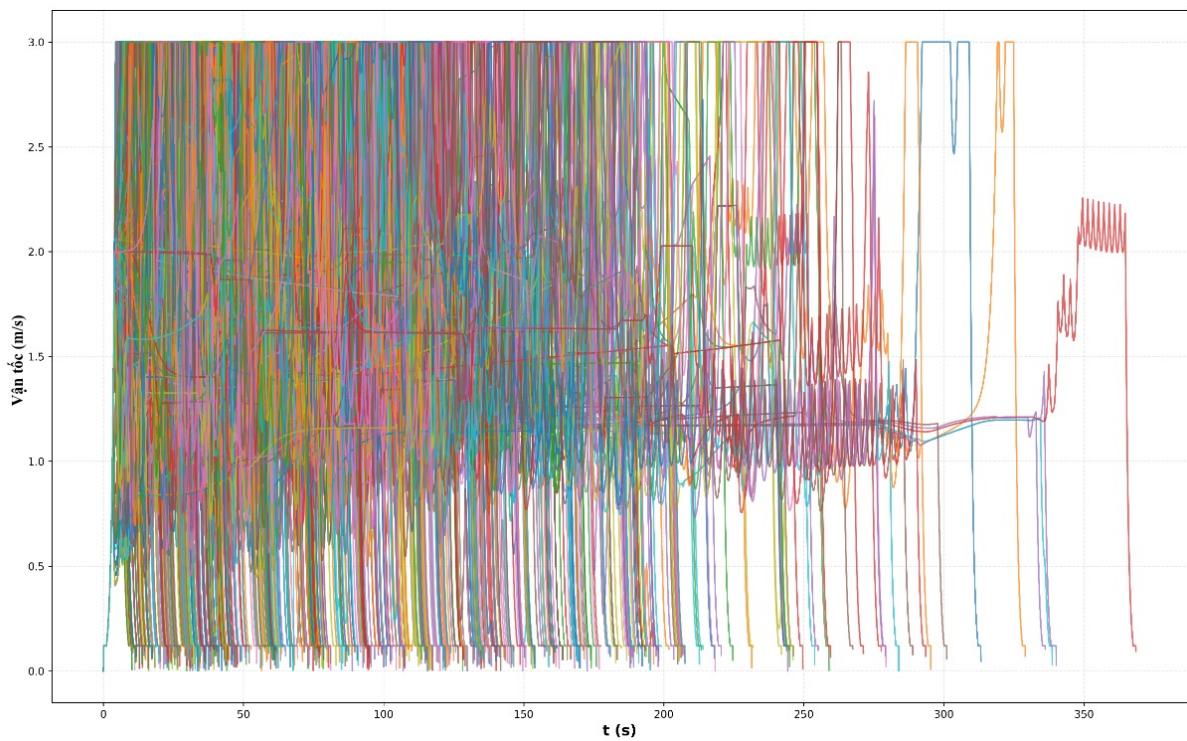
Với các tham số của thuật toán APF được lựa chọn như ở mục 4.2.1 cũng như kịch bản khảo sát như trên có thể chứng minh được sự hiệu quả của việc tích hợp thuật toán phân công nhiệm vụ bay vào thuật toán quy hoạch đường đi qua so sánh với phương pháp của Sun cùng cộng sự [10]. Trong nghiên cứu của Sun đã được công bố [10], việc phân công nhiệm vụ bay được đơn giản hóa theo cách dựa hoàn toàn trên chỉ số (drone 1 sẽ được phân công với mục tiêu 1), không xem xét đến tối ưu toàn cục. Mặt khác thuật toán hungarian luôn phân công drone với một mục tiêu tối ưu thông qua việc tối ưu hóa hàm tổng chi phí với chi phí được tính dựa trên khoảng cách Euclidean. Kết quả tối ưu này góp phần tạo ra các chuyến bay ngắn và ổn định hơn từ đó thời gian chuyển đổi đội hình sẽ được giảm đáng kể. Hơn nữa, sự kết hợp này làm giảm tần suất xảy ra các sự kiện trao đổi mục tiêu, vì các đường bay được tạo ra phù hợp hơn và có ít giao cắt hơn. Do đó, các drone ít có khả năng bị mắc kẹt trong trường thế của những drone khác đã đến đích.

Các kết quả so sánh được trình bày trong bảng 4.2, hình 4.3 và hình 4.4 cho thấy rõ hiệu quả của phương pháp Hungarian-APF được đề xuất so với phương pháp APF truyền thống. Về định lượng, chi phí gán trung bình đã giảm đáng kể từ 161.5 xuống 21.25, cho thấy việc tối ưu hóa toàn cục do thuật toán Hungarian mang lại đã giúp tối thiểu hóa tổng quãng đường bay giữa các drone và mục tiêu tương ứng. Do đó, tổng thời gian bay giảm từ 368.56 s xuống 43.8 s, tương ứng với mức cải thiện hiệu quả vận hành khoảng 8.4 lần. Ngoài ra, số lượng sự kiện trao đổi mục tiêu giảm từ năm xuống còn một, khẳng định rằng khung phương pháp được đề xuất đã giảm tần suất xuất hiện cực tiểu cục bộ trong các đội hình drone dày đặc.

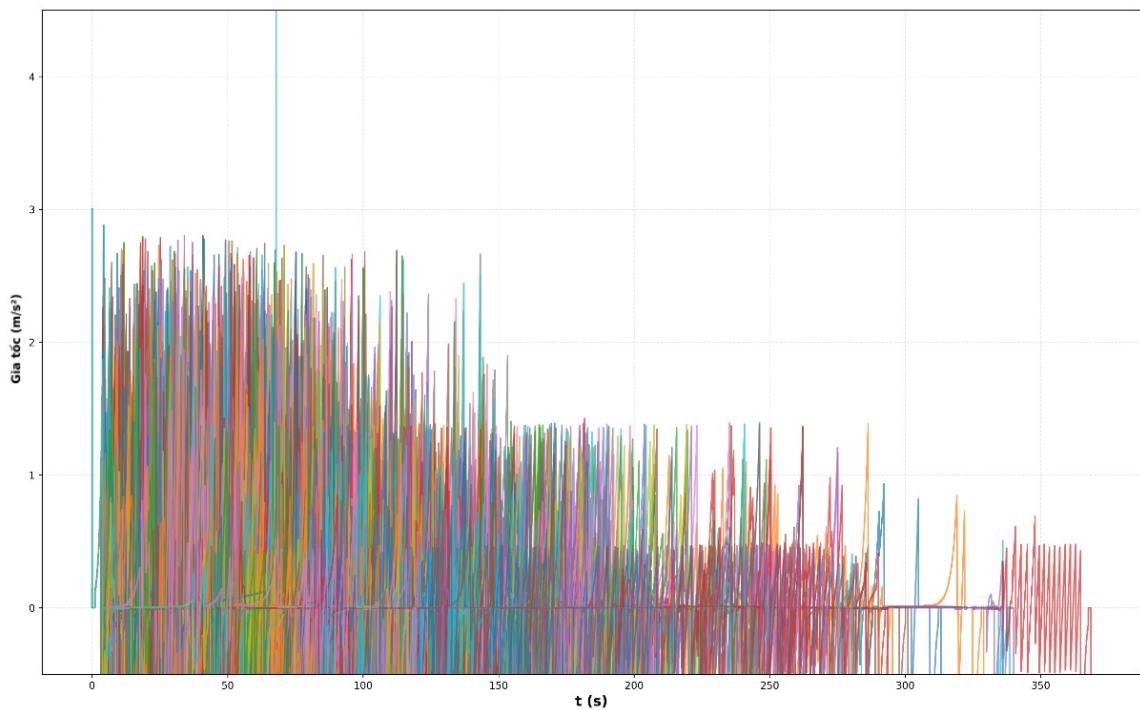
Về hiệu quả động học, gia tốc cực đại đã giảm từ 4.3 m/s^2 xuống 3.0 m/s^2 , trong khi vận tốc cực đại vẫn được giới hạn ở mức 3.0 m/s do các ràng buộc động học đã được thiết lập trước. Sự giảm thiểu các biến thiên gia tốc này cho thấy quỹ đạo trở nên mượt hơn và độ ổn định điều khiển được cải thiện. Khoảng cách tối thiểu giữa các drone tăng nhẹ từ 2.7 m lên 2.92 m, góp phần nâng cao độ an toàn của đội hình trong quá trình chuyển tiếp.

Bảng 4.2. Các thông số của buổi trình diễn

	APF	Hungarian kết hợp APF
Trung bình chi phí phân công	161.5	21.25
Thời gian thực hiện chuyển đổi	368.56	43.8
Số lần hoán đổi	5	1
Gia tốc lớn nhất ($\frac{\text{m}}{\text{s}^2}$)	4.1	3
Vận tốc lớn nhất ($\frac{\text{m}}{\text{s}}$)	3	3
Khoảng cách nhỏ nhất giữa các drone	2.7	2.82



Hình 4.11. Kết quả của vận tốc của phương pháp APF [10]



Hình 4.12. Kết quả của gia tốc phương pháp APF [10]

Chương 5.

Kết luận và hướng phát triển

Đồ án đã nghiên cứu và phát triển một khung chương trình tính toán điểm điều hướng cho các drone trong hệ thống trình diễn ánh sáng dựa trên phương pháp trường thế năng nhân tạo được cải tiến và tích hợp bài toán phân công nhiệm vụ được giải quyết bằng thuật toán Hungarian. Sự kết hợp hai phương pháp này đã mang lại hiệu quả rõ rệt trong quá trình chuyển đổi đội hình drone như giảm thời gian bay, giảm mắc kẹt cục bộ hay giúp các drone có các đường bay mượt mà hơn giảm sự thay đổi đột兀 vận tốc và gia tốc trong quá trình bay với thuật toán Hungarian giúp tổng quãng đường bay được tối thiểu hóa, thuật toán APF cải tiến giúp các drone tránh va chạm và tuân thủ các ràng buộc động học. Bên cạnh đó đồ án cũng triển khai thành công việc tăng tốc thuật toán quy hoạch đường thông qua lập trình song song trên GPU sử dụng PyOpenCL với phần tính toán lực tổng hợp được chọn là thành phần được song song hóa. Kết quả tính toán với các kịch bản có số lượng drone khác nhau đã chứng minh được hiệu quả của việc tăng tốc trên GPU so với việc thực hiện tuần tự trên CPU đặc biệt với số lượng drone lớn.

Mặt khác đồ án này vẫn còn nhiều mặt hạn chế cần được khắc phục như các tính toán được thực hiện chỉ dựa trên điều kiện lý tưởng với mỗi drone được mô hình như một vật rắn không xét đến các yếu tố khác như ảnh hưởng của gió, sai số định vị của GNSS-RTK và các cảm biến khác cũng như sai số của hệ thống điều khiển. Các tham số của thuật toán APF được tìm kiếm qua quá trình thử sai chỉ đảm bảo được tính an toàn chưa áp dụng các thuật toán tối ưu hóa để đảm bảo tìm được bộ tham số tối ưu cho phương pháp. Việc tăng tốc thuật toán quy hoạch đường đi chỉ tập trung vào phần tính toán lực tổng hợp chưa khai thác hết tiềm năng của việc lập trình song song trên GPU. Phần cứng tính toán sử dụng trong đồ án là card đồ họa gaming của AMD chưa phải là phần cứng chuyên dụng cho tính toán song song nên hiệu suất đạt được chưa cao.

Định hướng phát triển tương lai có thể tập trung vào các vấn đề như sử dụng các thuật toán tối ưu hóa để tìm bộ tham số tối ưu cho thuật toán APF cải tiến nhằm nâng cao chất lượng của quá trình chuyển đổi đội hình. Mô hình hóa các yếu tố thực tế như ảnh hưởng của gió, sai số định vị và điều khiển vào quá trình mô phỏng để đánh giá độ tin cậy của phương pháp trong điều kiện thực tế. Các thành phần có khả năng song song hóa khác như như thành phần tính toán bước nhảy động hay cập nhật vị trí ở bước lặp tiếp theo có thể được triển khai trên GPU để tận dụng tốt hơn sức mạnh tính toán song song. Sử dụng các phần cứng chuyên dụng cho tính toán song song như GPU dòng Tesla của Nvidia hay các cụm máy tính để nâng cao hiệu suất tính toán và có thể xem xét đến việc triển khai thời gian thực trên trạm mặt đất nếu thời gian tính toán đủ tốt.

Tài liệu tham khảo

Tiếng Việt

Tiếng Anh

- [1] Intel Corporation, "Intel Drone Light Show Breaks Guinness World Records Title at Olympic Winter Games PyeongChang 2018". [Online]. Available: <https://www.intc.com/news-events/press-releases/detail/172/intel-drone-light-show-breaks-guinness-world-records-title>.
- [2] Ehang, "EHang Egret's 1374 drones dancing over the City Wall of Xi'an, achieving a Guinness World Records title". [Online]. Available: <https://www.ehang.com/news/365.html>.
- [3] Masakazu Senda, "Vietnam marks 50th Reunification Day by lighting sky with record display of 10,500 drones". [Online]. Available: <https://www.guinnessworldrecords.com/news/commercial/2025/5/vietnam-marks-50th-reunification-day-by-lighting-sky-with-record-display-of-10500-drones>.
- [4] Huang, Jie, Guoqing Tian, Jiancheng Zhang, and Yutao Chen, "On Unmanned Aerial Vehicles Light Show Systems: Algorithms, Software and Hardware" *Applied Sciences*, 2021, Vol 11, no. 16.
- [5] CollMot Robotics Ltd., "CollMot multi drone show spec tech". [Online]. Available: <https://collmot.com/user/pages/resources/CollMot%20multi-drone%20show%20tech%20spec.pdf>.
- [6] Shenzhen DAMODA Intelligent Control Technology Co., Ltd, "Light show Drone L3". [Online]. Available: <https://www.damoda.com/products/l3.html>
- [7] Eric M.Frew, Timothy X. Brown, "Airbone Communication Networks for Small Unmanned Aircraft Systems", *Proceedings of the IEEEs*, 2008, Vol 96, Issue 12.
- [8] Wang, H.Zhao, Jiao Zhang, Dongtang Ma, JiaXun Li, Jibo Wei, "Survey on unmanned aerial vehicle networks: A cypher physical system perspective", *IEEE Communications Surveys & Tutorial*, 2019, Vol 22, Issue: 2.
- [9] Sin and Jun, "The dronetic moment: future of drone light show & lighting design in concerts", *Proceedings of IASDR 2023: Life-changing Design*, 2023.
- [10] H.Sun, J.Qi, M.Wang "Path Planning for Dense Drone Formation Based on Modiefield Aritificial Potential Field", *Proceedings of 39th Chinese Control Conference, Shenyang, China*, 2020, pp.4658-4664.
- [11] H. Chen, J. Qi, M. Wang, C. Wu, Y. Liu, and Y. Ping, "A Hybrid GA-APF 3D Trajectory Planning Method for Large-Scale UAV Formation Systems", *Proceedings*

of 42th Chinese Control Conference, 2023, pp.6024-6030.

- [12] G. E. Jan, T. Lei, C. -C. Sun, Z. -Y. You and C. Luo, "On the Problems of Drone Formation and Light Shows", *IEEE Transactions on Consumer Electronics*, 2024 , vol. 70, no. 3, pp. 5259-5268.
- [13] D. Nar and R. Kotecha, "Optimal Waypoint Assignment for Designing Drone Light Show Formations", *Results Control Optimal*, 2022, vol. 9, p. 100174.
- [14] A. Bakdi, A. Hentmout, H. Boutami, A. Maoudj, O. Hachor, B. Bourizia . "Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control". *Robot and Autonomous Systems*, 2017, 68, 1-29.
- [15] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, 1985.
- [16] NVIDIA Corporation, "NVIDIA CUDA C++ Programming Guide", 2023. [Online]. Available: <https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>.
- [17] Khronos Group, "The OpenCL Specification", 2023. [Online]. Available: https://www.khronos.org/registry/OpenCL/specs/3.0-unified/html/OpenCL_API.html.