



# Tính toán tối ưu quá trình chuyển đổi đội hình trong trình diễn ánh sáng

Vũ Đức Lương  
Giảng viên hướng dẫn: TS. Nguyễn Hoàng Quân

Viện Công Nghệ Hàng Không Vũ Trụ

15/12/2025



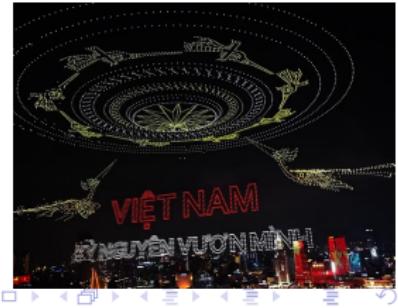
## Nội dung đồ án

- ▶ Tổng quan
- ▶ Cơ sở lý thuyết
- ▶ Triển khai thuật toán quy hoạch đường đi trên GPU
- ▶ Mô phỏng và phân tích kết quả
- ▶ Kết luận và hướng phát triển

# Tổng quan

## Giới thiệu

- ▶ Trong những năm gần đây, trình diễn ánh sáng bằng drone đã trở thành xu hướng mới trong các sự kiện lớn. Công nghệ này sử dụng các UAV bay theo đội hình để tạo hiệu ứng ánh sáng ấn tượng, có tính linh hoạt cao, an toàn và thân thiện với môi trường, dần thay thế pháo hoa truyền thống.
- ▶ Công nghệ này thu hút sự quan tâm của các công ty lớn như Intel, Ehang, Damoda với các màn trình diễn quy mô lớn.



# Tổng quan

## Phần cứng drone trình diễn

- ▶ Drone trình diễn ánh sáng thường sử dụng quadrotor do chi phí thấp và cấu trúc linh hoạt, gồm bộ điều khiển trung tâm tích hợp LED, khung sợi carbon và bốn cụm động cơ [1].
- ▶ Drone không thể đổi hướng và tốc độ tức thì; quỹ đạo bay phải tuân theo giới hạn vận tốc và gia tốc để đảm bảo an toàn và tính thẩm mỹ.



Thông số	Collmot [2]	Damoda L3 [3]
Loại	Quadrotor	Quadrotor
Trọng lượng	< 2 kg	530 g
Kích thước	Ø < 100 cm, cao ~30 cm	Ø 320 mm, cao 115 mm
Thời gian bay	10–12 phút	20–25 phút
Tốc độ tối đa	6 m/s	10 m/s
Độ cao tối đa	120 m	120 m
Khoảng cách tối thiểu	7 m	1.4 m

[1] 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.

[2] CollMot Robotics Ltd., "CollMot multi drone show spec tech".

[3] Shenzhen DAMODA Intelligent Control Technology Co., Ltd, "Light show Drone L3".

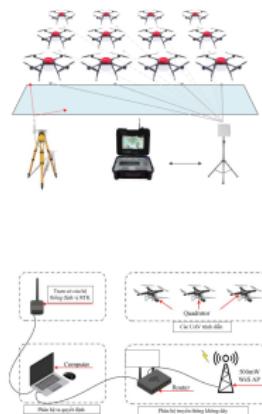
# Tổng quan

## Triển khai hệ thống trong thực tế

Triển khai các hệ thống drone trình diễn ánh sáng chế độ điều khiển bán tự động được sử dụng phổ biến do chế độ điều khiển tự động yêu cầu các cảm biến và khả năng tính toán của bộ điều khiển bay ở mỗi drone phải có độ chính xác cao. Trong khi đó chế độ điều khiển thủ công lại có tính rủi ro cao khi trình diễn số lượng drone quy mô lớn do có thể gặp lỗi truyền thông giữa các UAV và trạm mặt đất cũng như khả năng điều khiển của phi công.

[1] 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.

[4] H.Sun, J.Qi, M.Wang, "Path Planning for Dense Drone Formation Based on Modiefied Artificial Potential Field", Proceedings of 39th Chinese Control Conference, Shenyang, China, 2020, pp.4658-4664.





# Tổng quan

## Mục tiêu đồ án

Đồ án tập trung phát triển một công cụ giúp tính toán các waypoint các drone trong quá trình chuyển đổi đội hình với đầu vào là các đội hình được thiết kế sẵn dựa trên hai thuật toán:

- ▶ Thuật toán Hungarian giúp các drone được gán vào các mục tiêu tối ưu sao cho tổng quãng đường đi là nhỏ nhất.
- ▶ Thuật toán trường thế năng nhân tạo đã được cải tiến giúp các drone không va chạm trong quá trình chuyển đổi cũng như đảm bảo các yêu cầu về mặt động học. Ngoài ra thành phần tính toán lực tổng hợp của thuật toán cũng sẽ được triển khai tính toán trên GPU thông qua PyOpenCL giúp tăng tốc thuật toán quy hoạch đường đi trong trường hợp mô phỏng với số lượng drone lớn.

Các thuật toán được thực hiện theo hướng lập kế hoạch offline (tính toán từ trước sau đó triển khai) và được kiểm chứng thông qua mô phỏng.

[4] H.Sun, J.Qi, M.Wang, "Path Planning for Dense Drone Formation Based on Modiefied Artificial Potential Field", Proceedings of 39th Chinese Control Conference, Shenyang, China, 2020, pp.4658-4664.

[5] D. Nar and R. Kotecha, "Optimal Waypoint Assignment for Designing Drone Light Show Formations", Results Control Optimal, 2022, vol. 9, p. 100174.



# Cơ sở lý thuyết

## Mô tả bài toán

Trong không gian 3D, mỗi drone được xem như là một vật rắn và vị trí của nó được đại diện bởi

$$\mathbf{p}_i = (x_i, y_i, z_i).$$

Vị trí đội hình bắt đầu:

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

Vị trí đội hình tại thời điểm  $t$ :

$$\mathbf{P}_t = [\mathbf{p}_1^t, \mathbf{p}_2^t, \dots, \mathbf{p}_{n-1}^t, \mathbf{p}_n^t] \quad (2)$$

Vị trí đội hình mục tiêu:

$$\mathbf{P}_e = [\mathbf{p}_1^e, \mathbf{p}_2^e, \dots, \mathbf{p}_{n-1}^e, \mathbf{p}_n^e] \quad (3)$$

Ma trận phân công nhiệm vụ  $S$ .

$$S = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,n-1} & s_{1,n} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,n-1} & s_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ s_{n-1,1} & s_{n-1,2} & \cdots & s_{n-1,n-1} & s_{n-1,n} \\ s_{n,1} & s_{n,2} & \cdots & s_{n,n-1} & s_{n,n} \end{bmatrix} \quad (4)$$

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  $j$ .

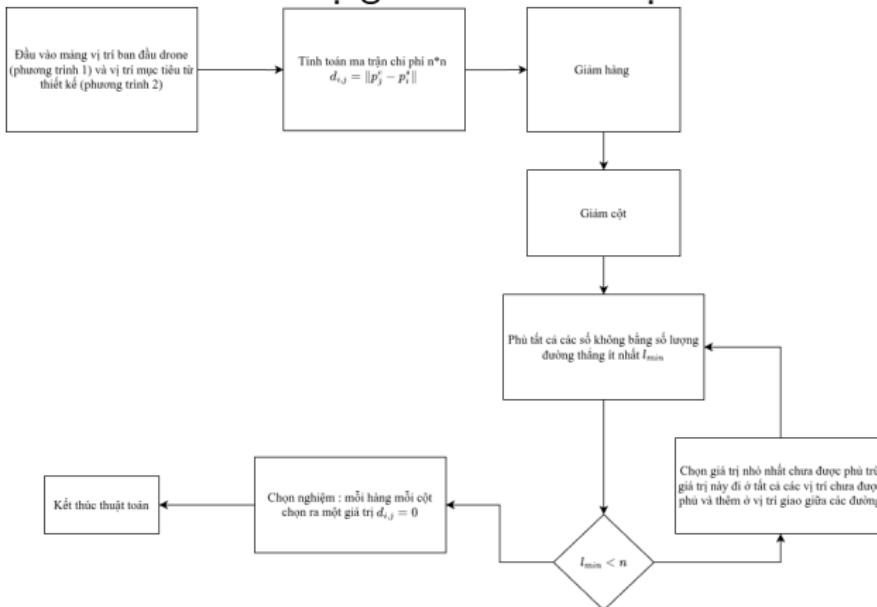
$$\min \sum_{i=1}^n \sum_{j=1}^n s_{i,j} d_{i,j} \quad (5)$$

$$\left\{ \begin{array}{l} \sum_{j=1}^n s_{i,j} = 1, \quad 1 \leq i \leq n \\ \sum_{i=1}^n s_{i,j} = 1, \quad 1 \leq j \leq n \end{array} \right. \quad (6)$$

# Cơ sở lý thuyết

## Áp dụng thuật toán Hungarian

Thuật toán Hungarian có độ phức tạp  $O(n^3)$  được áp dụng với đầu vào là mảng vị trí bắt đầu các drone và vị trí các mục tiêu từ thiết kế và đầu ra là ánh xạ giữa drone và mục tiêu





# Cơ sở lý thuyết

## Thuật toán trường thê năng nhân tạo

### Trường thê hấp dẫn

$$U_{att_i} = \frac{1}{2} \varepsilon \| \mathbf{p}_i^e - \mathbf{p}_i^t \|^2 \quad (7)$$

### Trường thê đẩy

$$U_{rep_{i,O}} = \begin{cases} \frac{1}{2} \eta \left( \frac{1}{\| \mathbf{p}_i^t - \mathbf{p}_O^t \|} - \frac{1}{\rho} \right)^2, & \text{nếu } \| \mathbf{p}_i^t - \mathbf{p}_O^t \| \leq \rho, \\ 0, & \text{nếu } \| \mathbf{p}_i^t - \mathbf{p}_O^t \| > \rho. \end{cases} \quad (8)$$

### Trường thê tổng hợp

$$U_{res_i} = U_{att_i} + \sum_{O=1}^m U_{rep_{i,O}} \quad (9)$$

### Lực tổng hợp

$$\mathbf{F}_{res_i} = -\nabla U_{res_i}. \quad (10)$$

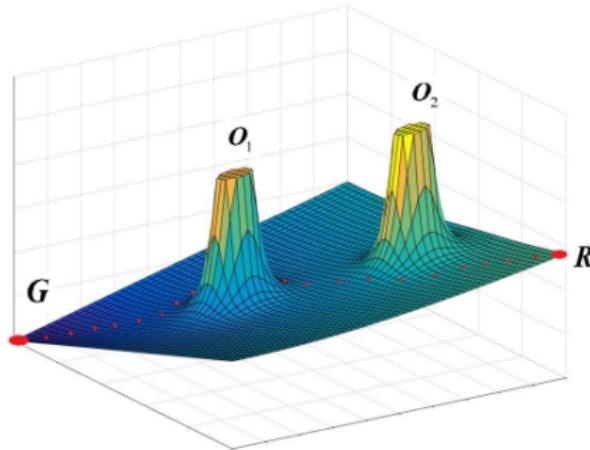
[6] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", Proceedings. 1985 IEEE International Conference on Robotics and Automation, 1985.

## Cơ sở lý thuyết

### Thuật toán trường thế năng nhân tạo

Sau khi tính toán được lực tổng hợp có thể tính toán các waypoint trong quá trình trình drone di chuyển ở bước lặp tiếp theo bằng cách dịch một khoảng  $\lambda$  theo hướng lực tính toán được

$$\mathbf{p}_i^{t+1} = \mathbf{p}_i^t + \lambda \cdot \frac{\mathbf{F}_{res_i}^t}{\|\mathbf{F}_{res_i}^t\|} \quad (11)$$



# Cơ sở lý thuyết

Thuật toán trường thê năng nhân tạo cải tiến cho drone  
trình diễn ánh sáng

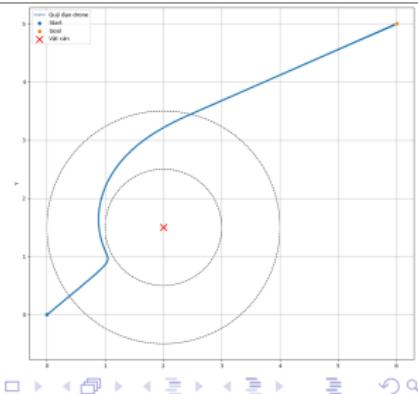
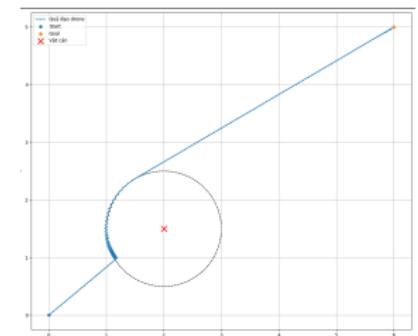
**Bước nhảy ràng buộc động học**

$$\lambda = \begin{cases} \sqrt{2ad_{\min}} \cdot T, & d_{\min} \leq \frac{v_{\max}^2}{2a} \\ v_{\max} \cdot T, & \text{ngược lại} \end{cases} \quad (12)$$

$$d_{\min} = \min(\|\mathbf{p}_i^s - \mathbf{p}_i^t\|, \|\mathbf{p}_i^e - \mathbf{p}_i^t\|, \|\mathbf{p}_o^t - \mathbf{p}_i^t\|) \quad (13)$$

**Trường thê đẩy phân lớp**

$$U_{\text{rep},o} = \begin{cases} \frac{1}{2}\eta_1 \left( \frac{1}{\|\mathbf{p}_o^t - \mathbf{p}_i^t\|} - \frac{1}{\rho_1} \right)^2, & \|\mathbf{p}_o^t - \mathbf{p}_i^t\| \leq \rho_1 \\ \frac{1}{2}\eta_2 \left( \frac{1}{\|\mathbf{p}_o^t - \mathbf{p}_i^t\|} - \frac{1}{\rho_2} \right)^2, & \rho_1 < \|\mathbf{p}_o^t - \mathbf{p}_i^t\| \leq \rho_2 \\ 0, & \|\mathbf{p}_o^t - \mathbf{p}_i^t\| > \rho_2 \end{cases} \quad (14)$$



# Cơ sở lý thuyết

Thuật toán trường thê năng nhân tạo cải tiến cho drone  
trình diễn ánh sáng

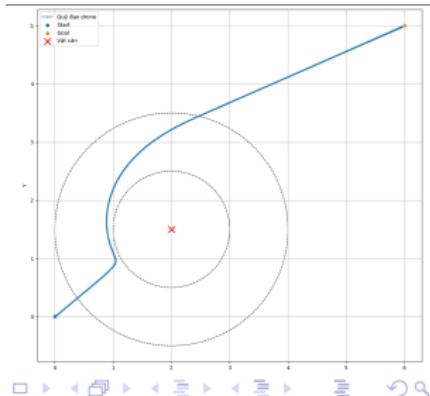
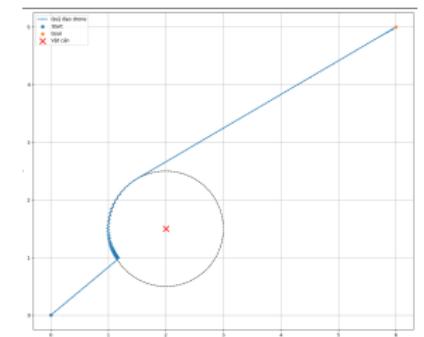
**Bước nhảy ràng buộc động học**

$$\lambda = \begin{cases} \sqrt{2ad_{\min}} \cdot T, & d_{\min} \leq \frac{v_{\max}^2}{2a} \\ v_{\max} \cdot T, & \text{ngược lại} \end{cases} \quad (15)$$

$$d_{\min} = \min(\|\mathbf{p}_i^s - \mathbf{p}_i^t\|, \|\mathbf{p}_i^e - \mathbf{p}_i^t\|, \|\mathbf{p}_o^t - \mathbf{p}_i^t\|) \quad (16)$$

**Trường thê đẩy phân lớp**

$$U_{\text{rep},o} = \begin{cases} \frac{1}{2}\eta_1 \left( \frac{1}{\|\mathbf{p}_o^t - \mathbf{p}_i^t\|} - \frac{1}{\rho_1} \right)^2, & \|\mathbf{p}_o^t - \mathbf{p}_i^t\| \leq \rho_1 \\ \frac{1}{2}\eta_2 \left( \frac{1}{\|\mathbf{p}_o^t - \mathbf{p}_i^t\|} - \frac{1}{\rho_2} \right)^2, & \rho_1 < \|\mathbf{p}_o^t - \mathbf{p}_i^t\| \leq \rho_2 \\ 0, & \|\mathbf{p}_o^t - \mathbf{p}_i^t\| > \rho_2 \end{cases} \quad (17)$$





## Cơ sở lý thuyết

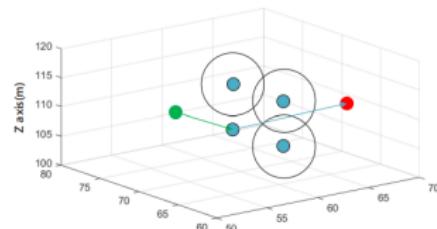
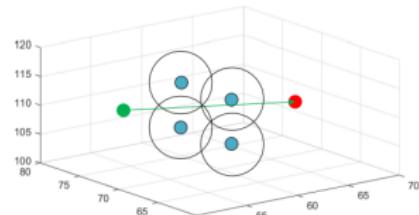
# Thuật toán trường thế năng nhân tạo cải tiến cho drone trình diễn ánh sáng

### Thuật toán hoán đổi mục tiêu

- ▶ 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.
- ▶ Điều kiện thứ hai có ít nhất một drone tạo trường đẩy giữa drone d và mục tiêu của nó.
- ▶ 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

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.

[4] H.Sun, J.Qi, M.Wang "Path Planning for Dense Drone Formation Based on Modiefield Artificial Potential Field", Proceedings of 39th Chinese Control Conference, Shenyang, China, 2020, pp.4658-4664.





# Cơ sở lý thuyết

## Khung chương trình tính toán điểm điều hướng đề xuất

$$\lambda_k = a_{max} \cdot T^2 \quad (18)$$

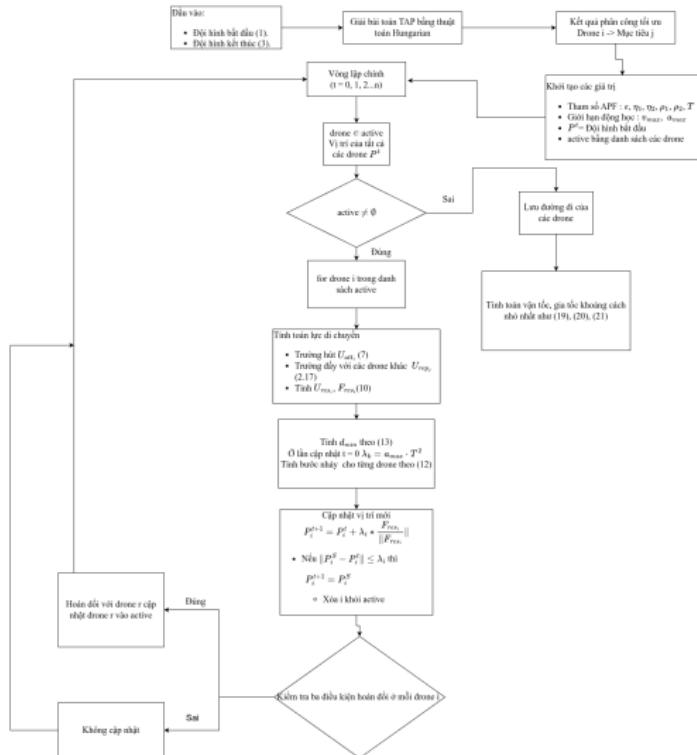
$$v_i^t = \|p_i^t - p_i^{t-1}\| \cdot T \quad (19)$$

$$a_i^t = \|a_i^t - a_i^{t-1}\| \cdot T \quad (20)$$

$$d_{min} = \min \|p_i^t - p_j^t\|,$$

$$j \neq i, i, j \in \{1, \dots, n\}$$

(21)





# Triển khai thuật toán quy hoạch đường đi trên GPU



## GPU và PyOpenCL

### Tại sao dùng GPU?

- Độ phức tạp thuật toán:  $O(n^2)$  với  $n$  drone
- CPU quá chậm khi mô phỏng hàng trăm drone
- GPU có hàng nghìn lõi xử lý song song

### Tại sao chọn PyOpenCL?

- Hỗ trợ đa nền tảng (AMD, NVIDIA, Intel)
- Card AMD Radeon RX 5600M (không hỗ trợ CUDA, ROCm)
- Giao diện Python dễ dùng, tích hợp NumPy phù hợp minh họa nguyên lý song song thuật toán

### Kiến trúc OpenCL:

- Platform:** Nhà cung cấp phần cứng
- Device:** GPU/CPU cụ thể
- Context:** Môi trường thực thi
- Kernel:** Hàm chạy song song
- Buffer:** Vùng nhớ chia sẻ

### Quy trình làm việc:

- Khởi tạo context và queue
- Tạo buffer cho dữ liệu
- Thực thi kernel trên GPU
- Đồng bộ và lấy kết quả



# Triển khai thuật toán quy hoạch đường đi trên GPU



## Lựa chọn thành phần song song hóa

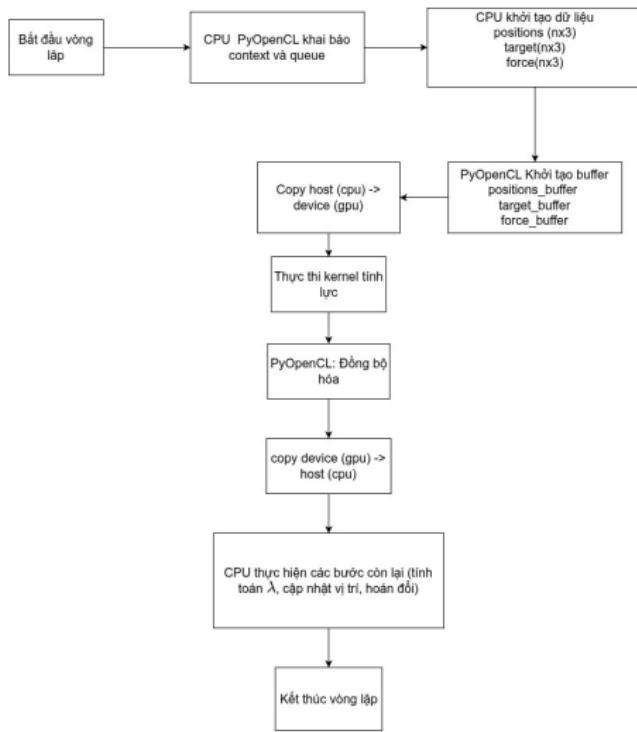
### Lựa chọn tính lực tổng hợp

Tính song song hóa cao

- ▶ Mỗi drone tính toán độc lập
- ▶ Cùng một công thức cho tất cả
- ▶ Ít rẽ nhánh, cấu trúc đều đặn

Chiếm phần lớn thời gian tính toán

- ▶ Độ phức tạp cao  $O(n^2)$
- ▶ Thực hiện nhiều phép tính toán so với phần tính  $\lambda$
- ▶ Luật Amdahl: tập trung tăng tốc phần tính toán lâu nhất để đạt hiệu quả tối đa





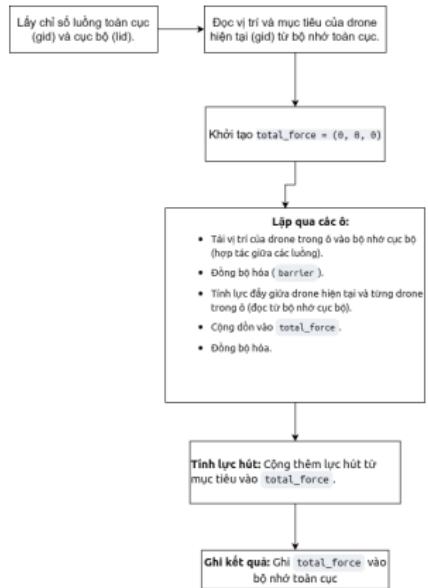
# Triển khai thuật toán quy hoạch đường đi trên GPU



## Kernel tính toán lực tổng hợp trên GPU

### Kiến trúc Song Song

- ▶ **Mô hình 1 drone - 1 luồng:** Ánh xạ trực tiếp, đơn giản hóa điều phối
- ▶ **Work-group 64 luồng:** Tận dụng tối đa kiến trúc GPU AMD (wavefront size = 64)
- ▶ **Kỹ thuật Chia Ô (Tiling):**
  - ▶ Giảm đáng kể truy cập bộ nhớ toàn cục (Global Memory)
  - ▶ Tăng tốc nhờ bộ nhớ cục bộ tốc độ cao (Local Memory)



[7] Cedric Nugteren, "Tutorial: OpenCL SGEMM tuning for Kepler"



# Mô phỏng và phân tích kết quả

## Kết quả thời gian tính toán

Mô phỏng được thực hiện trên máy tính cá nhân với CPU Intel Core i7-10750H (6 nhân, 12 luồng), RAM 16 GB và GPU AMD Radeon RX 5600M (6 GB).

- ▶ Với số lượng drone nhỏ (< 10), CPU nhanh hơn do chi phí khởi tạo GPU.
- ▶ Khi số lượng drone tăng, GPU thể hiện rõ ưu thế nhờ xử lý song song.
- ▶ Thời gian tính toán lực tổng hợp trên GPU dưới 0.5s ngay cả với số lượng drone lên đến 600 trong khi trên CPU tăng rất nhanh.

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

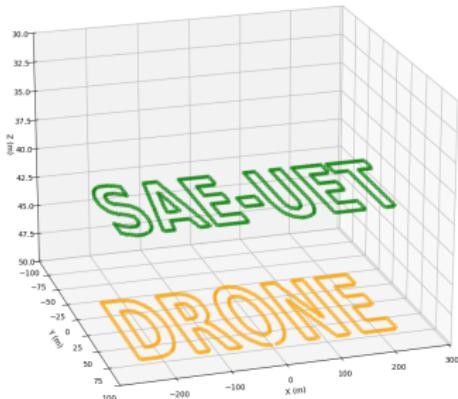
# Mô phỏng và phân tích kết quả

## Kịch bản khảo sát

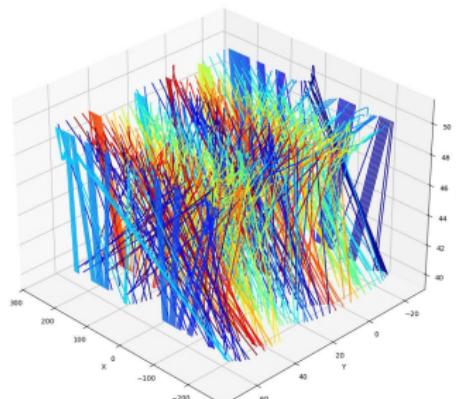
- ▶ Chuyển đổi đội hình từ chữ “SAE-UET” sang “DRONE” với 600 drone.
- ▶ Các thông số cố định:

$$\rho_1 = 2.5 \text{ m}, \quad \rho_2 = 3.5 \text{ m}, \quad a_{\max} = 3 \text{ m/s}^2, \quad v_{\max} = 3 \text{ m/s}, \quad T = \frac{1}{25} \text{ s}$$

- ▶ Khảo sát ảnh hưởng của các tham số APF: hệ số hấp dẫn  $\varepsilon$ , trường đẩy lớp trong  $\eta_1$  và lớp ngoài  $\eta_2$



Các đội hình thiết kế

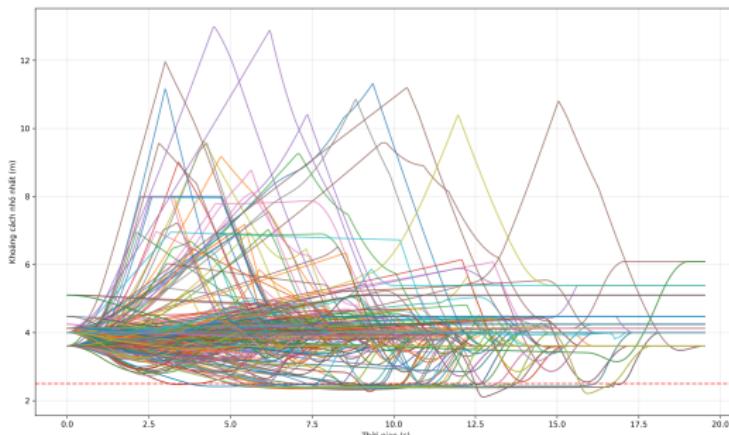
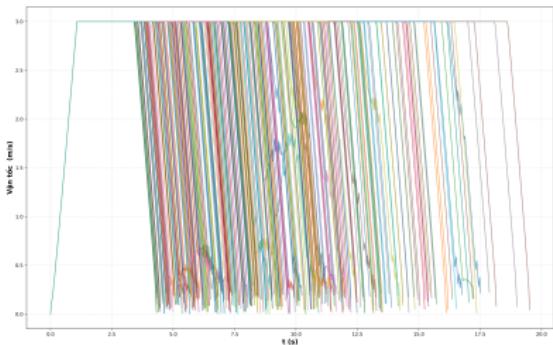
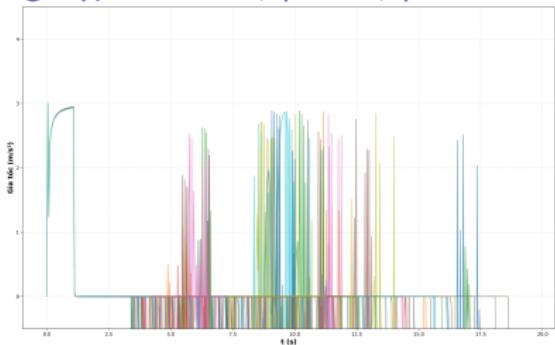


Đường đi cần phân tích



# Mô phỏng và phân tích kết quả

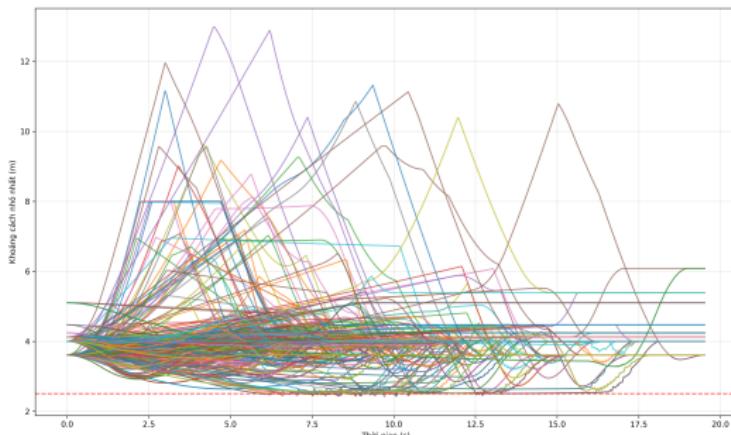
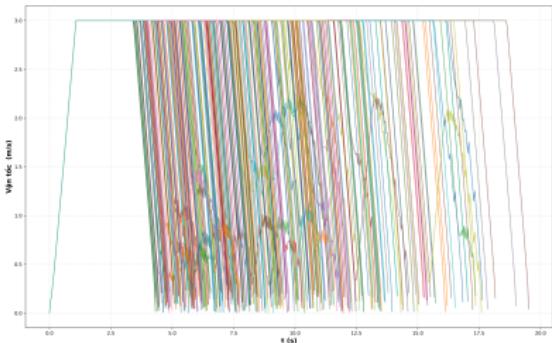
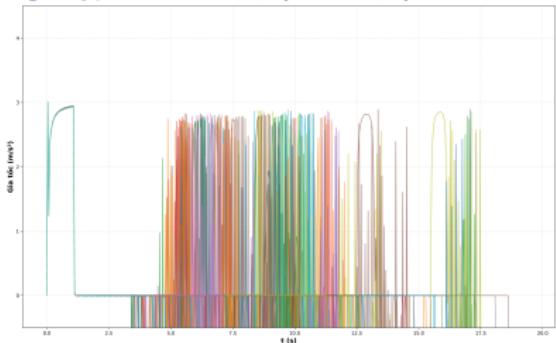
Trường hợp 1:  $\varepsilon = 5, \eta_1 = 5, \eta_2 = 500$





# Mô phỏng và phân tích kết quả

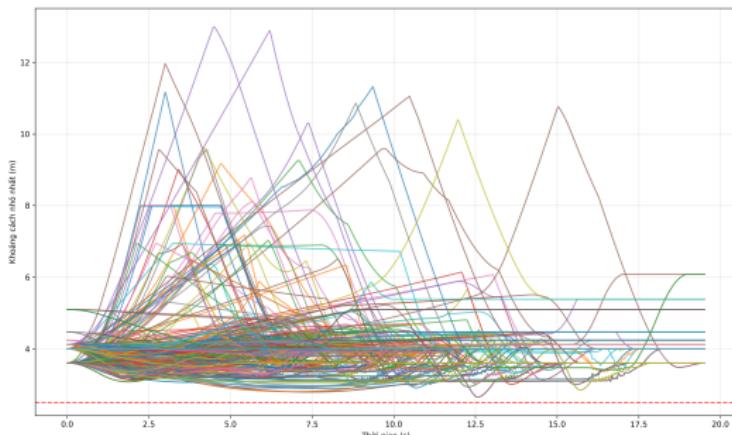
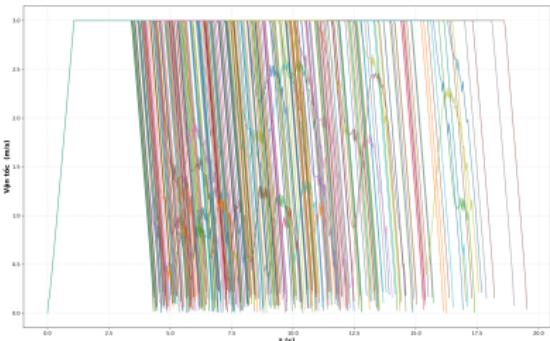
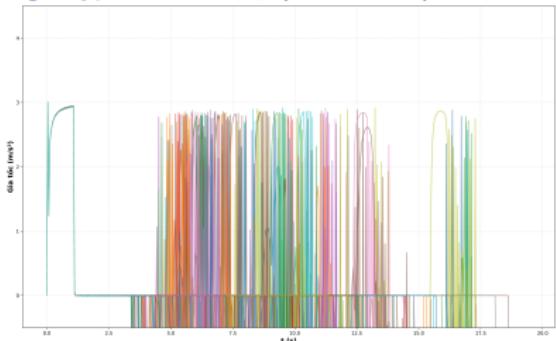
Trường hợp 2:  $\varepsilon = 3, \eta_1 = 8, \eta_2 = 500$





# Mô phỏng và phân tích kết quả

Trường hợp 3:  $\varepsilon = 3, \eta_1 = 30, \eta_2 = 500$





# Mô phỏng và phân tích kết quả

## So sánh với nghiên cứu của Sun [4]

- ▶ Phương pháp đề xuất giảm đáng kể thời gian chuyển đổi (19.03 lần).
- ▶ Giảm số lần hoán đổi, chứng tỏ ít xảy ra kẹt cục bộ.
- ▶ Quỹ đạo mượt hơn, an toàn hơn với khoảng cách giữa các drone được cải thiện, gia tốc được giới hạn tốt hơn trong khi đó vận tốc ở hai phương pháp luôn luôn được giới hạn.

Các thông số của buổi trình diễn

	APF	Hun-garian kết hợp APF
Trung bình chi phí phân công	161.5	21.25
Thời gian thực hiện chuyển đổi	368.56	19.36
Số lần hoán đổi	5	1
Gia tốc lớn nhất ( $\frac{m}{s^2}$ )	4.1	3
Vận tốc lớn nhất ( $\frac{m}{s}$ )	3	3
Khoảng cách nhỏ nhất giữa các drone	2.7	2.82

[4] 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.



# Kết luận và hướng phát triển

## Kết quả đạt được và hạn chế

### Kết quả đạt được

- ▶ Phát triển thành công khung tính toán điều hướng kết hợp thuật toán Hungarian giúp tối ưu phân công nhiệm vụ, giảm tổng quãng đường bay, trường thê năng nhân tạo cải tiến giúp tránh va chạm, đảm bảo ràng buộc động học và thoát khỏi cực tiểu cục bộ.
- ▶ Cải thiện hiệu suất chuyển đổi đội hình giảm thời gian bay và mắc kẹt cục bộ, khoảng cách các drone được đảm bảo hơn, ràng buộc động học tốt hơn
- ▶ Song song hóa phần tính lực tổng hợp với PyOpenCL đạt được hiệu quả rõ rệt với số lượng drone lớn.
- ▶ Một phần kết quả trong đồ án đã được trình bày tại **Vu Duc Luong, Tran Dang Huy, Nguyen Hoang Quan, "Assignment and Path Planning for Drone Light Show", 8<sup>th</sup> International Conference of Engineering Mechanics and Automation**

### Hạn chế

- ▶ Chưa xét ảnh hưởng của gió, sai số định vị GNSS-RTK và hệ thống điều khiển. Drone được mô hình hóa đơn giản như một vật rắn
- ▶ Tham số APF được chọn qua thử nghiệm chưa áp dụng tối ưu hóa tự động
- ▶ Phần tính toán GPU chỉ song song hóa phần tính lực phần cứng sử dụng là card gaming, chưa chuyên dụng cho tính toán hiệu năng cao.



# Kết luận và hướng phát triển

## Hướng phát triển

- ▶ Tích hợp ảnh hưởng môi trường như gió, nhiễu tín hiệu, hay mô hình bộ điều khiển bay thực tế.
- ▶ Áp dụng thuật toán meta-heuristic (GA, PSO) để tìm tham số APF tối ưu
- ▶ Khai thác tối đa hiệu năng GPU song song hóa thêm các thành phần tính, cập nhật vị trí và triển khai trên phần cứng chuyên dụng (NVIDIA Tesla)



# Cảm ơn thầy cô và các bạn đã lắng nghe bài trình bày

Mọi ý kiến đóng góp đều được trân trọng tiếp thu