Slide 1:

Kính thưa thầy giáo, cô giáo cùng toàn thể các bạn sinh viên, em là Vũ Trung Đức, sau đây em xin trình bày khóa luận tốt nghiệp về tài: Nghiên cứu mô phỏng hệ thống trình diễn ánh sáng bằng drone với kỹ thuật tối ưu quỹ đạo chuyển động.

Slide 2:

Phần trình bày của em bao gồm 4 phần:

* Giới thiệu chung nêu ra vấn đề, mục tiêu nghiên cứu.
* Cơ sở lý thuyết trình bày về:
  + Tổng quan hệ thống trình diễn ánh sáng UAV
  + Các thuật toán áp dụng trong khóa luận như Hungarian cho phân nhiệm, điều hướng tránh va chạm với APF, tối ưu chuyển động bằng MEC.
* Thiết kế hệ thống trình bày:
  + Kiến trúc tổng thể của một hệ thống UAV
  + Việc tích hợp thuật toán
  + Giao diện thiết kế kịch bản ADI
* Thực nghiệm và đánh giá:
  + So sánh hiệu quả giữa các phương pháp
  + Kết quả đạt được
  + Hướng phát triển trong tương lai

Slide 3:

Phần 1 giới thiệu chung

Slide 4:

Trong những năm gần đây, trình diễn ánh sáng drone dần thay thế pháo hoa truyền thống nhờ sự:

* an toàn (không gây cháy nổ).
* thân thiện với môi trường.
* Dễ dàng tái sử dụng
* Có thể kể đến như buổi trình diễn: rực rỡ thăng long năm 2025 hay nhiều màn trình diễn khác.
* Tuy nhiên việc triển khai thực tế yêu cầu đầu tư lớn: chi phí UAV, hệ thống định vị RTK, … kỹ thuật đồng bộ hóa UAV phức tạp. Đã có nhiều sự cố do nhiễu sóng, lỗi định vị hoặc mất điều khiển dẫn đến rơi hàng loạt UAV gây thiệt hại đáng kể: buổi trình diễn thử trước ngày 30/4-1/5
* Chính vì vậy việc mô phỏng trước khi thực hiện trở thành một đề tài đáng được chú ý.
* Câu hỏi đặt ra là:
  + Làm sao để xây dựng mô hình mô phỏng chính xác?
  + Làm sao để tối ưu quãng đường đi của drone là ngắn nhất?
  + Làm sao để đảm bảo các drone hạn chế va chạm trong khi di chuyển?

Ta cùng nhau đi tìm câu trả lời.

Slide 5:

Khóa luận có 4 mục tiêu chính:

* Xây dựng mô phỏng 3d trên Unity
* Tích hợp các thuật toán điều khiển
* Thiết kế giao diện người dùng trực quan
* Mô phỏng và kiển thử với 85 drone

Slide 6:

Phần 2: Giới thiệu chung

Slide 7:

Một hệ thống UAV thông thường gồm 2 phần chính:

Phần cứng: UAV nhỏ gọn, tích hợp đèn led, được định vị bằng cảm biến GNSS RTK, và nhận lệnh điều khiển từ trung tâm điều khiển mặt đất GCS.

Phần mềm: Trung tâm điều khiển (GCS) có chức năng lập kế hoạch trình diễn, phân nhiệm, điều hướng drone. Drone và GCS giao tiếp thông qua Wifi, MAVLink…

Slide 8:

Vậy làm sao để tối ưu cho quãng đường đi của done?

Bài toán phân nhiệm thực hiện tìm một ánh xạ 1-1 giữa độ hình ban đầu và đội hình mục tiêu sao cho tổng quãng đường bay của drone là ngắn nhất.

Slide 9:

Để thực hiện điều này, thuật toán Hungarian được sử dụng với đầu vào là ma trận chi phí giữa các UAV và các vị trí mục tiêu tương ứng. Đầu ra cho ra mỗi UAV cần di chuyển tới đích nào. Với độ phức tạp O(n^3).

Hỉnh ảnh minh họa có 3 người ứng với 3 công việc và chi phí mỗi người thực hiện từng công việc. Thuật toán Hungarian cho ra kết quả tối ưu.

Slide 10:

Thuật toán Hungarian bao gồm 5 bước chính:

Chuẩn hóa hàng: thực hiện trừ tất các phần tử trên cùng 1 hàng cho phần tử bé nhất hàng đó.

Chuẩn hóa cột: thay vì trừ trên cùng 1 hàng ta thực hiện trên cột.

Tìm phủ số 0 tối thiểu: thực hiện kiểm tra các hàng, hàng nào chứa số 0 ta gạch bỏ cột, sau khi kiểm tra các hàng xong, ta thực hiện kiểm tra các cột và gạch bỏ hàng tương ứng.

Nếu chưa đủ số lượng vị trí phân công ta thực hiện điều chỉnh ma trận: thực hiện lấy các phần tử chưa bị loại bỏ trong ma trận trừ đi phần tử bé nhất. Và lặp lại trên ma trận nhận được.

Ngược lại ta được phương án tối ưu.

Slide 11:

Em có minh họa các bước thực hiện thuật toán Hungarian:

Ma trận đầu vào lần lượt khoảng cách giữa drone và các vị trí mục tiêu

Sau bước chuẩn hóa hàng ta được ma trận mới

Sau bước chuẩn hóa cột ta được ma trận mới

Thực hiện tìm phủ số 0 tối thiểu: ở hàng 1 ta gạch cột 1, hàng 2, 3 ta gạch cột 4, hàng 4 không có, tiếp tục cột 1 không có, cột 2 ta gạch hàng 4, cột 3, 4 không có

Do chưa đủ 4 phần tử 0 được gạch ta thực hiện điều chỉnh ma trận. Cuối cùng ta được kết quả:

Slide 12:

Làm sao để đảm bảo các drone hạn chế va chạm trong khi di chuyển?

Thuật toán APF dựa trên nguyên lý lực đẩy và lực hút.

Giải thích trường thế

Slide 13

Tuy nhiên thuật toán APF luôn hoạt động tốt trong thực tế?

Đây là 2 trưởng hợp mà drone có thể gặp phải khi áp dụng APF.

TH1 drone bị chặn bởi hai drone khác và có tổng hợp lực đẩy có hướng ngược với lực hút

TH2: 2 drone di chuyển ngược nhau.

Slide 14:

Để giải quyết vấn đề này, ta sử dụng thêm tối ưu cục bộ. Khi drone rơi vào vùng nguy hiển, drone sẽ gửi thông báo tới trung tâm điều khiển, tại trung tâm điều khiển có trách nghiệm tìm kiếm các drone có trong phạm vi từ đó thực hiện phân nhiệm lại cho các drone này. Kết quả đạt được như hình dưới.

Slide 15:

Bên cạnh đó em có phát triển thêm việc tích hợp thuật toán MEC trong xử lý nén video. Khi một kịch bản động chứa nhiều frame liên tiếp nhau, nếu ta áp dụng Hungarian cho mỗi lần chuyển frame sẽ gây chậm hệ thống và lãng phí tài nguyên tính toán. Nguyên lý thuật toán sẽ dựa vào vị trí hiện tại và độ dịch chuyển với đích tiếp theo từ đó dịch chuyển tới mục tiêu. Vì vậy thay vì O(n^3) còn O(1)

Slide 16:

Thiết kế hệ thống

Slide 17:

Em đã thực hiện xây dựng hệ thống bao gồm 2 phần chính:

Phần thiết kế kịch bản trình diễn thực hiện xử lý dữ liệu kịch bản người dùng dựa trên dữ liệu có sẵn

Phần trình diễn ánh sáng: dữ liệu định dạng .csv được đưa tới khối Bộ quản lý thực hiện phân nhiệm và đưa ra yêu cầu với trình điều khiển drone, rồi truyền dữ liệu tới drone mô phỏng. Trong quá trình di chuyển, drone liên tục gửi trạng thái tới bộ quản lý đồng thời gửi dữ liệu tọa độ về để bộ quản lý phân tích sau này.

Slide 18:

Tại bộ quản lý thực hiện:

B1: Nhận dữ liệu từ modun ADI

B2: Đọc dữ liệu kịch bản, đánh dấu id cho từng drone

B3: Sử dụng tối ưu toàn cục bằng Hungarian.

Mỗi khi drone gửi thông báo hoàn thành nhiệm vụ, thực hiện kiểm tất cả các drone đã hoàn thành hay chưa nếu đã hoàn thành chuyển sang B4

B4: Thực hiện chuyển frame đối với kịch bản động, ngược lại quay lại B3 với kịch bản tiếp theo.

Slide 19:

Tại bộ điều khiển drone:

B1: Nhận dữ liệu từ bộ quản lý về đích đến

B2: Kiểm tra va chạm để xử lý cục bộ

B3: Di chuyển drone:

Thực hiện tính toán APF với frame đầu tiên

Thực hiện tính toán MEC với frame tiếp theo

B4: Khi drone tới vị trí đích sẽ thông báo tới bộ quản lý đồng thời hiển thị màu sắc tương ứng.

Slide 20:

Với mục tiêu dễ dàng thao tác, giao diện ADI có chức năng chính:

Thêm kịch bản

Xóa kịch bản

Sắp xếp thứ tự trình diễn

Slide 21:

Kết quả và đánh giá

Slide 22:

Trong quá trình thực hiện, em có xử lý 2 trường hợp:

Có sử dụng tối ưu cục bộ và không sử dụng trên 4 kịch bản: 2 kịch bản đầu có mật độ drone nhỏ, 2 hình sau có mật độ drone lớn và đánh giá qua nhiều lần thử nghiệm thu được kết quả trung bình như hình bên.

Trong đó đường màu xanh là khi đã tối ưu cục bộ, đường màu vàng là không tối ưu cục bộ.

Ta thấy rằng phương pháp không áp dụng tối ưu cục bộ có nhiều đoạn dài đưa khoảng cách xuống gần mức giới hạn nguy hiểm

59(2.0 đơn vị), trong khi phương pháp có áp dụng tối ưu cục bộ duy trì khoảng cách ổn định và an toàn hơn trong phần lớn thời gian bên cạnh đó thời gian thực hiện của cũng giảm đáng kể (giảm 10 giây trên tổng số 65 giây so với việc không áp dụng).

Biểu đồ bên phải thể hiện số lượng cặp drone rơi vào vùng nguy hiểm (< 2m) theo thời gian. Phương pháp không tối ưu thường xuất hiện các đỉnh cao và rải rác, cho thấy nhiều thời điểm xảy ra hiện tượng các drone tiến quá gần nhau – tăng rủi ro va chạm. Trong khi đó, phương pháp tối ưu cục bộ giúp giữ số lượng cặp nguy hiểm ở mức thấp, thậm chí bằng 0 trong phần lớn chu kỳ mô phỏng

Tổng quãng đường và đánh giá hiệu quả:

*−* Trung bình mỗi drone di chuyển khoảng 153.81m khi áp dụng tối ưu cục bộ,

so với 154.22m khi không áp dụng.

*−* Tổng quãng đường toàn đội hình đạt 13073.68m (có tối ưu) và 13108.64m

(không tối ưu)

Slide 23:

Kết quả đạt được

- Thực hiện mô phỏng với 85 drone với nhiều kịch bản

- Áp dụng các thuật toán vào hệ thống

- Thực hiện thử nghiệm và đánh giá

Slide 24:

Hướng phát triển tương lai

Hiện tại drone chưa có hệ thống tính toán năng lượng tiêu hao, chính vì vậy việc tích hợp mô hình năng lượng là điều cần thiết.

Bên cạnh đó việc áp dụng trí tuệ nhân tạo giúp drone tự xác định được quỹ đạo là một hướng đi mới.

Thêm vào đó để có thể trình diễn nhiều loại kịch bản, quy mô cần được cải tiến.

Slide 25:

Em xin chân thành cảm ơn Quý Thầy Cô và Hội đồng đã lắng nghe phần trình bày của em. Đây không chỉ là một đề tài kỹ thuật, mà còn mang tính ứng dụng thực tế cao.

Em rất mong nhận được các góp ý quý báu từ Thầy Cô để hoàn thiện đề tài tốt hơn trong tương lai. Bên cạnh đó em có tích hợp chương trình trên web thầy cô và các bạn có thể quét mã để truy cập.

**Em xin trân trọng cảm ơn!**

Câu hỏi có thể nhận:

❓Câu 1: Vì sao em chọn Unity thay vì Gazebo hay Webots?

Dạ, em chọn Unity vì đây là một nền tảng mạnh về đồ họa 3D và tương thích tốt với C# – ngôn ngữ lập trình em sử dụng để tích hợp các thuật toán.

So với Gazebo hay Webots thiên về mô phỏng vật lý robot, Unity giúp em:

Thiết kế giao diện ADI trực quan dễ dùng hơn,

Dễ dàng thêm hiệu ứng ánh sáng – màu sắc – chuyển động mượt,

Đồng thời em có thể dễ dàng xuất bản mô phỏng này lên web.

Do mục tiêu chính là trình diễn ánh sáng và kiểm tra thuật toán phân nhiệm + chuyển động đồng bộ, Unity là lựa chọn phù hợp hơn với đề tài.

❓Câu 2: Nếu drone thật không có GNSS RTK thì sao? Có thể mô phỏng không?

Dạ, nếu drone không có GNSS RTK thì độ chính xác vị trí sẽ giảm xuống còn khoảng vài mét, dẫn đến việc trình diễn sẽ dễ bị lệch đội hình, va chạm.

Tuy nhiên trong mô phỏng, em giả lập GNSS RTK bằng cách đưa vào tọa độ chính xác nên vẫn có thể đánh giá thuật toán tốt.

Nếu triển khai thật, có thể thay thế GNSS RTK bằng các công nghệ định vị tương đối như Ultra Wide Band (UWB) hoặc Visual SLAM trong phạm vi nhỏ.

Em cũng đề xuất trong hướng phát triển là mô phỏng sai số vị trí, để đánh giá độ ổn định thuật toán.

❓Câu 3: Nếu tăng số drone lên 500 thì hiệu năng hệ thống có đảm bảo?

Dạ, về mặt lý thuyết, hệ thống vẫn chạy được, tuy nhiên cần tối ưu một số điểm:

Thuật toán Hungarian có độ phức tạp 𝑂(𝑛^3), nên với 500 drone cần áp dụng thuật toán xấp xỉ (đưa ra các phương án gần tối ưu) hoặc chia vùng để xử lý song song.

Unity có thể mô phỏng hàng trăm object, nhưng cần giảm số frame hoặc tối ưu rendering để tránh lag.

MEC sẽ giúp giảm số lần phân nhiệm, giảm tải tính toán tại GCS.

❓Câu 4: MEC em triển khai có tương đương thuật toán Block Matching trong video không?

Dạ, về nguyên lý thì tương tự:

Cả hai đều dùng ý tưởng từ Motion Estimation để dự đoán vị trí tiếp theo từ frame hiện tại.

MEC em áp dụng là phiên bản đơn giản hơn: không tìm khối tương đồng mà lấy vector dịch chuyển trung bình giữa drone hiện tại và mục tiêu mới.

Điều này giúp giữ được hướng bay liên tục, giảm chấn động hoặc đột ngột khi chuyển đội hình, mà không cần phân nhiệm lại bằng Hungarian mỗi frame.

❓Câu 5: Em có đề xuất gì để đưa hệ thống từ mô phỏng sang thực tế?

Dạ, nếu triển khai thực tế, em đề xuất:

Tối ưu lại mô phỏng cho phần cứng thật, dùng drone mini như Tello để kiểm tra trước.

Tích hợp hệ thống định vị nội bộ, ví dụ UWB nếu không dùng được GNSS.

Giao tiếp với drone thật bằng MAVSDK thay cho Unity giả lập.

Phát triển thêm mô-đun kiểm soát năng lượng, tránh hết pin giữa chừng.

Cuối cùng là đưa giao diện ADI ra thành ứng dụng web thực tế để người dùng không kỹ thuật cũng có thể tạo kịch bản trình diễn.