

Đề tài: Triển khai giải thuật MPC với robot hai bánh vi sai trong bài toán bám quỹ đạo và tránh vật cản

Họ tên: Cao Huy Nhật

MHV: 21025118

1. Giới thiệu

Hiện nay, robot di động là một trong những lĩnh vực rộng lớn nhất lĩnh vực nghiên cứu và trở nên phổ biến hơn trong ngành công nghiệp. Robot di động có thể di chuyển tự động theo nhiều cách khác nhau môi trường. Trong môi trường trong nhà có nhiều chướng ngại vật, nơi không gian di chuyển của robot bị hạn chế. Và đó là yếu tố quan trọng để xây dựng việc điều hướng và tránh vật cản cho robot

Điều hướng là bài toán thiết yếu đối với robot tự hành, trong đó hệ thống điều khiển quỹ đạo đóng vai trò quyết định đến hành vi của robot và điều hướng robot di chuyển trong không gian. Điều khiển dự đoán mô hình phi tuyến (NMPC), được gọi là receding horizon control là một phương pháp miền thời gian dựa trên tối ưu hóa để điều khiển vòng kín của hệ thống phi tuyến. Ưu điểm chính của NMPC là nó có thể xử lý rõ ràng các ràng buộc hệ thống trong quy trình kiểm soát, điều thường bị bỏ qua trong các phương pháp truyền thống. NMPC tính toán các hành động điều khiển tại mỗi khoảng thời gian điều khiển bằng cách sử dụng kết hợp cơ sở mô hình dự đoán và tối ưu hóa có ràng buộc. Do đó, NMPC là một phương pháp thích hợp hơn khi giải quyết các vấn đề kỹ thuật khi có các ràng buộc của hệ thống. Các ràng buộc của hệ thống robot di động có thể đến từ vận tốc, gia tốc và khoảng cách tối đa của mỗi bánh xe đến các chướng ngại vật xung quanh để tránh va chạm.

Dựa trên khả năng dự đoán trạng thái tương lai của robot, bộ điều khiển NMPC đặc biệt phù hợp với các bài toán dẫn đường và tránh chướng ngại vật trong môi trường. Do đó, robot có thể xác định chướng ngại vật và tính toán các tín hiệu điều khiển tối ưu để di chuyển mượt mà hơn so với các thuật toán trước đây.

2. Cơ sở lý thuyết

2.1. Simultaneous localization and mapping (SLAM)

SLAM là viết tắt của Simultaneous localization and mapping. Nó đề cập đến thách thức xây dựng bản đồ môi trường xung quanh robot trong khi robot đang di chuyển trong thời gian thực mà không có bất kỳ kiến thức nào trước đó về môi trường. SLAM phục vụ hai mục đích. Tạo bản đồ môi trường của robot đồng thời theo dõi vị trí và quỹ đạo của robot.

Hiện nay có rất nhiều phương pháp để giải quyết bài toán SLAM, điển hình là Visual SLAM và Lidar SLAM. Trong bài toán này em sử dụng Lidar SLAM để thu thập thông tin môi trường và xây dựng bản đồ.

2.2. Dẫn đường (Nagivation)

Dẫn đường cho robot là một vấn đề quan trọng đặc biệt trong các bài toán của robot di động. Để một robot di động di chuyển trong môi trường một cách tự do, linh hoạt, robot cần các thông tin về môi trường đặc biệt là bản đồ của môi trường đó. Trong bài toán này, thông tin của bản đồ sử dụng là bản đồ sau của môi trường sau khi thực hiện giải thuật SLAM. Khi đã có bản đồ robot sẽ lập đường đi đến một điểm cố định trên bản đồ đó. Để lập đường đi, có rất nhiều giải thuật để lập đường đi như A*, RRT, D*... trong bài toán này em sử dụng A* để lập đường đi cho robot.

2.3. Model Predictive Control (MPC)

Ở giai đoạn tiếp theo, robot tiến hành di chuyển theo quỹ đạo chuyển động đã được lập. Tuy nhiên, từ những sai số hệ thống, tác động từ môi trường robot có thể không thực hiện bám theo quỹ đạo một cách tốt nhất, dẫn đến cần một giải thuật để tối ưu chuyển động và tránh vật cản ngẫu nhiên trong môi trường di chuyển của robot.

Model Predictive Control (MPC) là một cách tiếp cận đầy hứa hẹn cho điều chỉnh tối ưu các hệ thống phi tuyến tính. Mô hình điều khiển MPC yêu cầu một mô hình chính xác về động lực học của hệ thống và đòi hỏi nguồn tài nguyên tính toán đáng kể so với các khái niệm điều khiển thông thường như PID hoặc LQR (Linear quadratic regulators). Trong suốt hai thập kỷ qua, mối quan tâm đến việc triển khai nhanh chóng về mặt tính toán và hiệu quả về mặt số học của MPC đã tăng lên đáng kể trong cả nghiên cứu lý thuyết và nghiên cứu định hướng ứng dụng.

Mục tiêu của bộ điều khiển MPC là để tối thiểu hóa hàm chi phí trong một khoảng thời gian cụ thể bằng cách tối ưu các trạng thái ước lượng sau đó dựa trên trạng thái trước đó của robot.

3. Mô hình động học của robot

Để thực hiện việc điều khiển tối ưu chuyển động bám quỹ đạo và tránh va chạm cho robot trong môi trường dựa trên bản đồ hóa được xây dựng từ giải thuật SLAM thì em đã sử dụng mô hình robot điều khiển vi sai – TurtleBot 3.

Robot được coi là một chất điểm chuyển động trong đồ thị 2D. Tọa độ của robot i được xác định bởi ba thành phần $P(x, y, \theta)$. Vận tốc $\vec{V}(x_v, y_v, \theta_v)$ của robot i sẽ bằng vi phân tọa độ của P :

$$\begin{cases} v_x = \frac{\Delta x}{\Delta t} \\ v_y = \frac{\Delta y}{\Delta t} \\ v_\theta = \frac{\Delta \theta}{\Delta t} \end{cases}$$

Với tâm của robot được đặt ở chính giữa hai bánh của robot. $\vec{V}(x_V, y_V, \theta_V)$ là vector vận tốc mong muốn của robot, chúng ta có thể sử dụng một mô hình động học dành cho robot hai bánh vi sai. Có phương trình chuyển động của robot

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x + V \cos(\theta) \\ y + V \sin(\theta) \\ \omega \end{bmatrix}$$

Vận tốc bánh trái ω_L và bánh phải ω_R được tính như sau:

$$\omega_L = |\vec{V}| - \zeta * \frac{\theta_{t+1} - \theta_t}{\Delta t}$$

$$\omega_R = |\vec{V}| + \zeta * \frac{\theta_{t+1} - \theta_t}{\Delta t}$$

Với hằng số ζ là hệ số điều chỉnh cố định. θ_{t+1} , θ_t là góc giữa hướng của robot và trục Oy tại thời điểm $t + 1$ và t .

Mô hình TurtleBot3 trong thực tế:

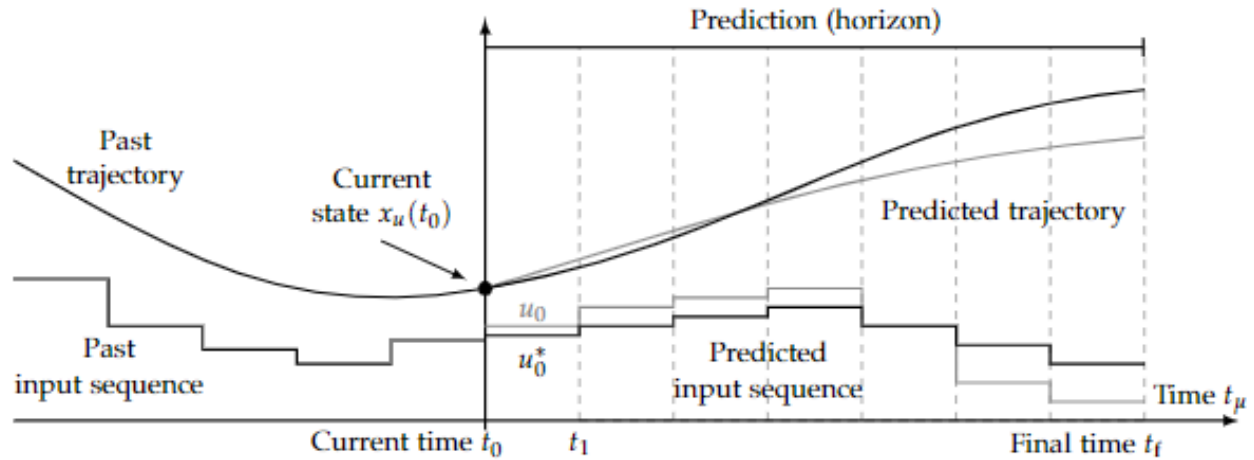


Dựa vào mô hình robot đã xây dựng, em tiến hành thực hiện giải thuật MPC để tiến hành nhiệm vụ di chuyển bám quỹ đạo và tránh vật cản cho robot trong dẫn đường dựa trên bản đồ được xây dựng từ giải thuật SLAM.

4. Giải thuật MPC

4.1. Model Predictive Control (MPC)

Mục tiêu của bộ điều khiển này là giảm thiểu hàm chi phí được xác định trước trong khi đáp ứng các ràng buộc như động lực học của hệ thống, giới hạn cơ cấu chấp hành, ... Tại mỗi time step, MPC sẽ tính toán một tập hợp các hành động (tín hiệu điều khiển) để giảm thiểu hàm chi phí trong một khoảng thời gian cụ thể và cho hành động cho time step tiếp theo, quy trình này sẽ lặp lại ở time step tiếp theo.



Giải thuật MPC tối ưu hơn so với các phương pháp điều khiển khác ở những đặc điểm. Thích hợp với những bài toán với nhiều ràng buộc từ môi trường như sai số từ thiết kế robot, sai số giữa các bánh xe hay tác động bởi vật cản ngẫu nhiên trong môi trường. Cũng như giải thuật đạt được hiệu quả cao. Tuy nhiên giải thuật có độ phức tạp cao và tốn nhiều thời gian tính toán.

Đầu vào của bộ điều khiển MPC là $v(t)$ và $x(t)$ lần lượt là vận tốc và vị trí của robot tại các thời điểm. Hệ thống điều khiển phi tuyến liên tục có thể được biểu thị bằng phương trình:

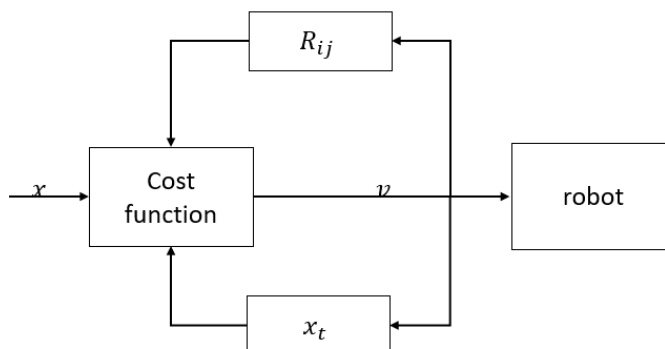
$$\dot{x}(t) = f(x(t), u(t))$$

Ở đây hàm f ở đây dựa trên mô hình động học của robot. Cách tiếp cận này để xác định vị trí của robot tại thời điểm thứ $t+1$ của robot tương đương với $x(t+1)$. Bộ điều khiển MPC được đưa ra nhằm tối thiểu hóa hàm chi phí để đưa ra tối ưu vị trí cho robot. Hàm chi phí ở đây được mô tả:

$$\min_{x,v} \sum_t^T C(x_t, v_t)$$

4.2. Thiết kế hệ thống điều khiển sử dụng bộ điều khiển MPC

Để áp dụng bộ điều khiển MPC cho bài toán bám quỹ đạo và tránh va chạm. Em xây dựng hệ thống điều khiển vòng kín cho giải thuật:



Hàm chi phí của MPC được xây dựng dựa trên việc tối ưu bám quỹ đạo với hai sai số khoảng cách giữa vị trí thực của robot và vị trí mong muốn d và hướng của robot và ϕ là đường tiếp tuyến của đường đi. Cùng với bám quỹ đạo robot cần tránh vật cản với $\zeta(r_a - r_{ij})$ là đại lượng thể hiện khoảng cách của robot với vật cản. Trong đó r_a là khoảng cách tối thiểu của robot với vật cản, r_{ij} là khoảng cách thực có chúng với ζ là đại lượng thể hiện sức ảnh hưởng về hướng lên robot. Dạng bậc hai chuẩn của hàm chi phí $C(x_t, v_t)$ mục tiêu của bài toán MPC có thể được định nghĩa như sau:

$$C(x_t, v_t) = \sum_t^T \{w_v ||v_t - v_t^{ref}||^2 + w_d ||d_t||^2 + w_\phi ||\phi_t||^2\} - \sum_j w_{ij} e^{\zeta(r_a - r_{ij})} \hat{r}_{ij}$$

Trong đó: w_v, w_d, w_ϕ là các trọng số biến đổi thể hiện các ràng buộc bám quỹ đạo của robot. Thay đổi các giá trị nhằm đưa robot về với quỹ đạo được định sẵn. Cùng với đó em xây dựng hàm tránh va chạm hướng đến việc robot tránh vật cản mà không cần tìm lại tuyến đường mới. Với j là các vị trí của các vật cản mà cảm biến thu được và chúng nằm trong bán kính ra. w_{ij} mô phỏng vị trí tác động của vật cản trong môi trường tại các vị trí khác nhau.

5. Kết quả

Thiết kế turtlebot 3 trong thực tế tương đương với mô hình mô phỏng. Em thực hiện kiểm chứng kết quả của giải thuật với hai bài thực nghiệm bám quỹ đạo và tránh vật cản tĩnh và vật cản động với các thông số của bộ điều khiển:

Đại lượng	Giá trị
$w_v(m/s^2)$	50
$w_d(m)$	50
$w_\phi(rad)$	1000
w_{ij}	1

ζ	10
$r_a(m)$	0.2

Kết quả của đề tài bao gồm 2 video:

https://drive.google.com/drive/folders/1LecwiRcJEsWRInWiB9JVaqzefjISwVq2?usp=share_link

6. Kết luận

Trong tiểu luận này em đã nghiên cứu thử nghiệm bộ điều khiển MPC cho bài toán bám quỹ đạo và tránh vật cản cho robot trong bài toán dẫn đường trên mô hình robot turtlebot 3. Kết quả cho thấy hệ thống thực hiện hoàn thành yêu cầu của bài toán với việc bám quỹ đạo di chuyển đến đích cũng như tránh vật cản tĩnh và vật cản động trong môi trường.

Tuy nhiên đáp ứng của hệ thống vẫn chưa thực sự cao về mặt thời gian tính toán do độ phức tạp của thuật toán đặc biệt là trong các môi trường có độ phức tạp cao như vật cản động.

Trong nghiên cứu tiếp theo, em sẽ tiến hành tối ưu thuật toán, ứng dụng các kỹ thuật tối ưu nhằm điều chỉnh các tham số trong quá trình di chuyển cũng như giảm thời gian đáp ứng của hệ thống.