



NGHIÊN CỨU, THIẾT KẾ, CHẾ TẠO ROBOT HỖ TRỢ NHÂN VIÊN Y TẾ TRONG VIỆC KHỬ KHUẨN LAU SÀN NHÀ

Lê Duy Tùng, Đỗ Trọng Tấn*, Nguyễn Bá Đạt,
Nguyễn Việt Cảnh, Phạm Đình Kha, Lê Bình Dương

Trung tâm Công nghệ Vi điện tử và Tin học, Viện Ứng dụng Công nghệ

* Tác giả liên hệ: dttan@most.gov.vn

Ngày tòa soạn nhận được bài báo: 08/10/2020

Ngày phản biện đánh giá và sửa chữa: 26/11/2020

Ngày bài báo được duyệt đăng: 28/01/2021

Tóm tắt:

Bài báo trình bày về chế tạo robot hỗ trợ y tế, kết hợp với các giải thuật SLAM để định vị và xây dựng bản đồ số, đồng thời thuật toán D^* Lite được sử dụng để tính toán, tối ưu hóa quỹ đạo di chuyển của robot. Robot có với tính năng di chuyển tự động, tránh vật cản, lựa chọn đường đi ngắn nhất, tự động sạc và thực hiện các tác vụ hỗ trợ y tế. Vật liệu chế tạo robot phù hợp với môi trường y tế. Hệ thống phun khử khuẩn và cơ cấu lau sàn giúp đảm bảo tính năng hỗ trợ nhân viên y tế trong việc khử khuẩn và lau sàn nhà.

Từ khóa: Robot hỗ trợ y tế, giải thuật SLAM, thuật toán D^* Lite.

1. Đặt vấn đề

Trong hơn nửa thế kỷ qua, ngành robotic đã có những bước phát triển và tiến hóa mạnh mẽ. Nhờ sự tối ưu về phần cứng như: các thiết bị cảm biến và vi xử lý có tốc độ, độ chính xác ngày càng cao trong khi giá thành giảm, công nghệ viễn thông và định vị cực kỳ phát triển và đã ứng dụng vào dân sự, robot đang ngày càng thay thế con người trong rất nhiều công việc. Theo dự báo [1] thì đến năm 2025, mỗi người sẽ có nhu cầu sử dụng robot cá nhân như cần một máy tính PC hiện nay và robot sẽ trở thành một cuộc cách mạng công nghệ lớn sau Internet. Cũng chính vì điều này, một lượng lớn các nghiên cứu về robot đã được công bố, tập trung chủ yếu vào việc ứng dụng robot trong các mặt của đời sống. Có thể kể đến một số các nghiên cứu về các loại robot đang được quan tâm hiện nay như: Tay máy robot (Robot Manipulators) [2], robot di động (Mobile Robots) [3], robot mô phỏng sinh học (Bio Inspired Robots) [4] và robot cá nhân (Personal Robots) [5]. Vấn đề chủ yếu được quan tâm là các thuật toán hỗ trợ robot phục vụ con người.

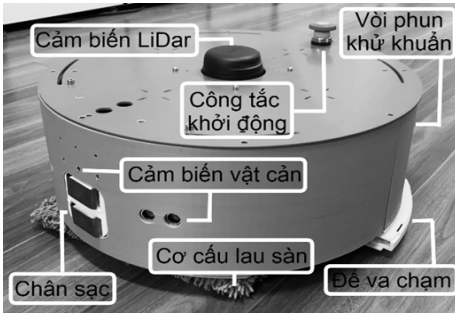
Trong bài báo này, chúng tôi nghiên cứu, thiết kế và chế tạo robot hỗ trợ y tế (Robot-assisted Medical) thuộc dòng robot di động với tính năng chính là lau sàn và khử khuẩn tự động. Giải thuật

SLAM [6] được sử dụng để xử lý bài toán xây dựng môi trường và định vị, thuật toán D^* Lite [7] xử lý bài toán tìm đường. Nội dung nghiên cứu được trình bày dưới đây và được thể hiện qua các kết quả đáng tin cậy.

2. Nội dung nghiên cứu

2.1. Thiết kế chế tạo robot

Với mục tiêu hỗ trợ y tế, robot được thiết kế với khung ngoài bằng nhựa PLA (không độc), dạng tròn giúp dễ di chuyển, tối ưu về góc cạnh và không gian lắp đặt bên trong. Cấu tạo bên ngoài bao gồm các bộ phận: Cảm biến LiDar, chân sạc, công tắc khởi động, vòi phun khử khuẩn, cảm biến vật cản, cơ cấu lau sàn và đế va chạm (Hình 2.1). Trong đó, robot hoạt động dựa trên sự “quan sát” của cảm biến LiDar và các cảm biến vật cản hỗ trợ. Phía trước robot có đế va chạm là công tắc hành trình, giúp robot khi va chạm vào các chướng ngại vật thì sẽ đổi hướng chuyển động. Phía sau robot có bộ chân sạc, khi robot có tín hiệu cạn pin thì sẽ tự động chạy về điểm sạc và kết nối với đế sạc để sạc pin. Đồng thời, thực hiện chức năng khử khuẩn với 2 vòi phun khử khuẩn ở mặt trước và cơ cấu lau sàn ở mặt sau robot.



Hình 2.1. Hình ảnh bên ngoài robot



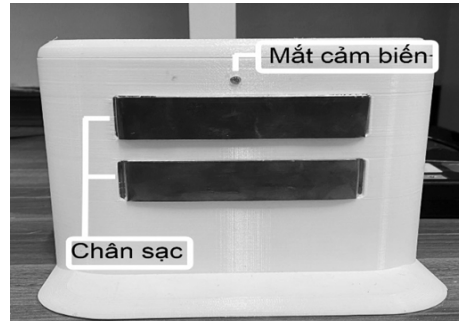
Hình 2.2. Sơ đồ khối phần cứng robot

Bên trong robot bao gồm 2 khối chính: khối mạch điện và khối khử khuẩn (Hình 2.2). Trong đó, khối mạch điện bao gồm: mạch cảm biến (xử lý tín hiệu từ cảm biến LiDar, cảm biến vật cản, la bàn điện tử, cảm biến đo mức dung dịch, mức pin), mạch nguồn (cung cấp nguồn cho động cơ và bơm), mạch điều khiển (điều khiển bơm và động cơ, mạch sạc ac quy), mạch giao tiếp với máy tính nhúng và ac quy. Khối khử khuẩn bao gồm bình đựng dung dịch khử khuẩn, bơm, hệ thống vòi và béc phun. Toàn bộ khối mạch điện và khối khử khuẩn được đặt gọn bên trong robot như trên Hình 2.2.

Phần mạch điện của robot bao gồm nhiều module mạch khác nhau. Tuy nhiên, tất cả các thông tin về cảm biến và bộ điều khiển đều được đưa về máy tính nhúng để thực hiện tính toán, xử lý dữ liệu và ra lệnh điều khiển. Do số lượng các module khá nhiều nên thiết kế chi tiết của module sẽ không được thể hiện trong bài báo này.

Phần cuối cùng là đế sạc cho robot (Hình 2.3)

với cấu trúc sạc trực tiếp thông qua tiếp xúc giữa hai điện cực trên robot và trên điểm sạc. Để được tích hợp cảm biến hồng ngoại để xác định vị trí tương đối với robot, đảm bảo việc di chuyển chính xác vào vị trí sạc. Trên thân đế sạc được tích hợp mạch sạc 24 V - 2 A và được gắn cố định tại điểm sạc. Thời lượng sạc cho robot từ 20% dung lượng ac quy đến 100% dung lượng ac quy là 48 phút. Tuy nhiên, do hoạt động trong diện tích nhỏ (phòng bệnh cỡ 25m²) nên luôn đảm bảo hoàn thành chu trình làm việc.



Hình 2.3. Đế sạc robot

2.2. Giải thuật SLAM

Bài toán đầu tiên trong việc điều khiển robot là định vị và xây dựng môi trường. Để có được vị trí của robot thì cần phải kết hợp 2 yếu tố đó là cảm biến (LiDar) và giải thuật xử lý tín hiệu và số hóa môi trường (SLAM). Trên thực tế, có 2 phương pháp tiêu biểu cho bài toán này là: Đo lường thị giác (Visual Odometry, Fast Appearance based Mapping) [8] và Định vị và lập bản đồ thời gian thực (SLAM) [6]. Trong bài báo này, chúng tôi sử dụng giải thuật SLAM do sự phù hợp với thiết bị phần cứng và tính thời gian thực của nó. Cụ thể hơn, chúng tôi sử dụng kỹ thuật FastSLAM [9] dựa trên bộ lọc nhiễu hạt Rao-Blackwellized [10] để xử lý dữ liệu sau cảm biến và xây dựng bản đồ số. Kết quả của kỹ thuật FastSLAM là ma trận hậu nghiệm Y_t bao gồm $[x_t^{[k]}, (\mu_{j,t}^k, \Sigma_{j,t}^k), \dots, (\mu_{j,t}^k, \Sigma_{j,t}^k)]$ với $k = 1 : M$ hạt và $j = 1 : N$ đặc trưng, $x_t^{[k]}$ là trạng thái của hạt thứ k tại thời điểm t . Nội dung cụ thể của giải thuật được trình bày như sau:

Đầu tiên, lấy $M (k = 1 : M)$ hạt và các đặc trưng của nó $[x_{t-1}^{[k]}, (\mu_{1,t-1}^k, \Sigma_{1,t-1}^k), \dots, (\mu_{j,t-1}^k, \Sigma_{j,t-1}^k)]$ từ ma trận hậu nghiệm Y_{t-1} , với $x_{t-1}^{[k]} = [x_{x(t-1)}^{[k]}, x_{y(t-1)}^{[k]}, \theta_{t-1}^{[k]}]^T$ là trạng thái thứ k tại thời điểm t .

Bước thứ hai, ta quan sát các đặc trưng mới và thêm chúng vào tập trạng thái. Ta thêm đại lượng đặc trưng chính $\mu_{j,t}^k$ và hiệp phương sai $\Sigma_{j,t}^k$ vào vector của các hạt theo phương trình:

$$\begin{bmatrix} \mu_{j,t}^k \end{bmatrix} = \begin{bmatrix} \mu_{x,j,t}^k \\ \mu_{y,j,t}^k \end{bmatrix} = \begin{bmatrix} x_{xt}^{[k]} + r \cos(\theta_t^{[k]} + \phi) \\ x_{yt}^{[k]} + r \sin(\theta_t^{[k]} + \phi) \end{bmatrix} \quad (1)$$

với r và ϕ là khoảng cách và góc lệch đo được. Để xác định ma trận Jacobian của (1) với khoảng cách và góc tương ứng, ta có:

$$H_m = \begin{bmatrix} \cos(\theta_t^{[k]} + \phi) & -r \sin(\theta_t^{[k]} + \phi) \\ \sin(\theta_t^{[k]} + \phi) & r \cos(\theta_t^{[k]} + \phi) \end{bmatrix} \quad (2)$$

Ma trận hiệp phương sai của đặc trưng cũng có thể tính bằng:

$$\Sigma_{j,t}^k = H_m Q_t H_m^T \quad (3)$$

với Q_t là ma trận hiệp phương sai của nhiễu đo.

Bước thứ ba, khi robot quan sát lại một đặc trưng cũ (giả sử đã có dữ liệu từ bước trước), ta chỉ cần cập nhật lại các thông số quan trọng của hạt. Chúng ta cần tính toán các giá trị trạng thái dựa trên vị trí của hạt sử dụng:

$$\hat{z} = \begin{bmatrix} \sqrt{(\mu_{x,j,t}^k - x_{xt}^{[k]})^2 + (\mu_{y,j,t}^k - x_{yt}^{[k]})^2} \\ \arctan\left(\frac{\mu_{y,j,t}^k - x_{yt}^{[k]}}{\mu_{x,j,t}^k - x_{xt}^{[k]}}\right) - \theta_t^{[k]} \end{bmatrix} \quad (4)$$

Tiếp theo, ta tính ma trận Jacobian của (4) đối với đặc trưng sử dụng:

$$H = \begin{bmatrix} \frac{\mu_{x,j,t}^k - x_{xt}^{[k]}}{r} & \frac{\mu_{y,j,t}^k - x_{yt}^{[k]}}{r} \\ -\left(\frac{\mu_{y,j,t}^k - x_{yt}^{[k]}}{r^2}\right) & \frac{\mu_{x,j,t}^k - x_{xt}^{[k]}}{r^2} \end{bmatrix} \quad (5)$$

$$r = \sqrt{(\mu_{x,j,t}^k - x_{xt}^{[k]})^2 + (\mu_{y,j,t}^k - x_{yt}^{[k]})^2} \quad (6)$$

Sau đó, chúng ta tính ma trận hiệp phương sai và hệ số Kalman:

$$Q = H \Sigma_{j,t-1}^k H^T + Q_t \quad (7)$$

$$K = \Sigma_{j,t-1}^k H^T Q^{-1} \quad (8)$$

và cập nhật đặc trưng chính và hiệp phương sai:

$$\mu_{j,t}^k = \mu_{j,t-1}^k + K(z_t - \hat{z}) \quad (9)$$

$$\Sigma_{j,t}^k = (I - KH) \Sigma_{j,t-1}^k \quad (10)$$

Cuối cùng, trọng số $w^{[k]}$ [10], được tính bởi:

$$w^{[k]} = |2\pi Q|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(z_t - \hat{z})^T Q^{-1}(z_t - \hat{z})\right) \quad (11)$$

với các trạng thái chưa được quan sát, ước lượng về vị trí của chúng không đổi. Giải thuật này sẽ được chạy liên tục cho đến khi hết M hạt. Tùy vào lựa chọn số lượng hạt chúng ta sẽ có sự cân đối giữa tốc độ giải thuật và độ chính xác của giải thuật.

Trong phiên bản robot này, khi thực hiện xây dựng bản đồ số, robot thực hiện chạy một lần xung quanh khu vực hành lang bệnh viện và đưa ra kết quả như Hình 2.4. Bản đồ sau đó sẽ được chia thành dạng lưới với các ô có kích thước là 20x20 cm (trong thực tế, xấp xỉ với một nửa kích thước mặt ngoài của robot) và thực hiện định vị trên bản đồ dạng lưới này.



Hình 2.4. Kết quả bản đồ số

2.3. Thuật toán D* Lite

Thuật toán D* Lite [7] được Sven Koenig và Maxim Likhachev đề xuất vào năm 2002. Thuật toán có ưu điểm có thể sử dụng các thông tin của tuyến đường đi trong lịch sử để nhanh chóng cập nhật và đưa ra tuyến đường mới khi có những yếu tố mới xuất hiện trong quá trình di chuyển (vật cản, sự cố, ...). Thuật toán bắt đầu đánh giá hàm chi phí từ các điểm lân cận đến điểm đích rồi dò ngược trở về để đưa ra con đường ngắn nhất từ điểm xuất phát đến điểm đích. Trước khi tiến hành, chia bản đồ số thu được từ giải thuật FastSLAM ở bước trên thành các ô có diện tích 20x20 cm. Chúng tôi định nghĩa các ký hiệu trong thuật toán như sau:

Bảng 2.1. Ký hiệu trong thuật toán D* Lite

STT	Ký hiệu	Ý nghĩa
1	X, Y	Trạng thái của robot
2	c(X, Y)	Chi phí di chuyển thực từ Y đến X
3	r(X, Y)	Chi phí di chuyển từ Y đến X dựa trên cảm biến
4	t(X)	Thông báo trạng thái của X (NEW, OPEN, CLOSED)

5	$h(X)$	Chi phí di chuyển
6	$k(X)$	Giá trị nhỏ nhất của $h(X)$
7	$g(X)$	Giá trị ước tính của hàm mục tiêu
8	$rhs(X)$	Ước tính một bước của giá trị hàm mục tiêu

Trạng thái của các ô trống như sau:

- Nếu $g = rhs$ thì ô đó ở trạng thái “consistent”.
- Nếu $g > rhs$ thì ô đó ở trạng thái “over-consistent”.
- Nếu $g < rhs$ thì ô đó ở trạng thái “under-consistent”.

Nếu có một tuyến đường trực tiếp nối hai điểm a, b trên bản đồ thì khi đó, b được gọi là “successor” của a và a được gọi là “predecessor” của b. Giá trị rhs của một ô được tính dựa trên giá trị của g. Ban đầu, giá trị g của ô đích là 0.

$$rhs(u) = \min_{U' \in U(u)} (c(u, s') + g(s')) \quad (12)$$

Sự ưu tiên của một ô được xác định:

$$\left[\min(g(s), rhs(s)) + h(s_{start}, s); \min(g(s), rhs(s)) \right] \quad (13)$$

Tiếp theo là các bước triển khai thuật toán:

Ban đầu, rhs và g của tất cả các ô được đặt bằng ∞ và OPEN list được để trống. Sau đó, giá trị g của ô đích được đặt bằng 0 và giá trị rhs được đặt bằng 0. Ô đích được đưa vào OPEN list và cờ sẽ được gắn ở ô đích.

Ta xét hai trường hợp xảy ra:

- Nếu ô chứa cờ có trạng thái over-consistent quá nhất quán thì giá trị g của ô đó được gắn bằng giá trị rhs ($g := rhs$). Sau đó tất cả các ô liền kề được đưa vào OPEN list và giá trị rhs của các ô này được cập nhật. Tiếp theo, ô chứa cờ được đưa ra khỏi OPEN list.

- Nếu ô chứa cờ có trạng thái under-consistent, giá trị g của nó được đặt bằng ∞ . Sau đó tất cả các ô liền kề được đưa vào OPEN list và giá trị rhs của các ô này được cập nhật. Tiếp theo, ô chứa cờ được đưa ra khỏi OPEN list. Chọn ô tiếp theo để xác định mức độ ưu tiên của các ô trong OPEN list.

Nếu robot phát hiện vật cản, giá trị rhs của ô chứa vật cản được đặt bằng ∞ . Chi phí của các ô liền kề ô chứa vật cản được đặt bằng ∞ , vì vậy con đường đang xét qua ô chứa vật cản sẽ bị loại bỏ.

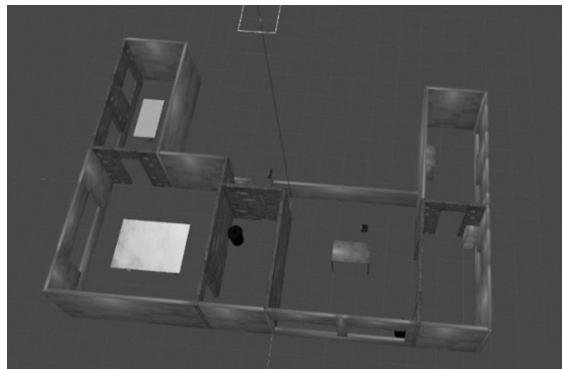
Lặp lại bước 2 và 3 cho đến khi trạng thái ô chứa cờ giống với trạng thái ô xuất phát hoặc không

có đường đi nào tồn tại. Như vậy, chúng ta sẽ có được con đường đi ngắn nhất.

Thuật toán áp dụng vào robot, kết hợp với thông số thu được từ các cảm biến vật cản giúp tối ưu hơn cho thuật toán tìm đường và hỗ trợ cho việc di chuyển của robot trở nên mượt mà hơn. Các ứng dụng cụ thể được trình bày trong phần tiếp theo.

3. Thử nghiệm và đánh giá

Để đánh giá tính đúng đắn của thuật toán và khả năng hoạt động của thuật toán, nhóm tác giả thực hiện hai kịch bản thử nghiệm cho robot trong điều kiện thực tế và trong mô phỏng với công cụ Gazebo (công cụ khởi tạo môi trường hoạt động), Rviz (công cụ giả lập thông tin từ các cảm biến).



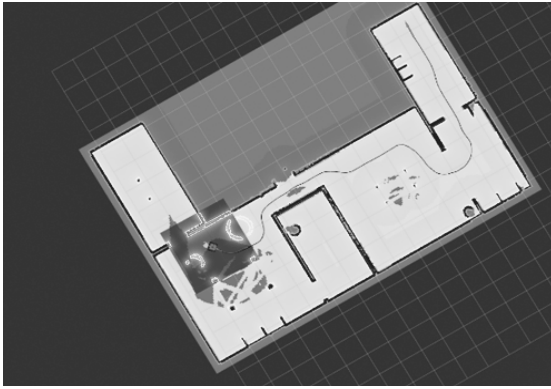
Hình 3.1. Môi trường thử nghiệm bằng Gazebo

Kết quả bản đồ số xây dựng được dựa trên kỹ thuật FastSLAM như sau:



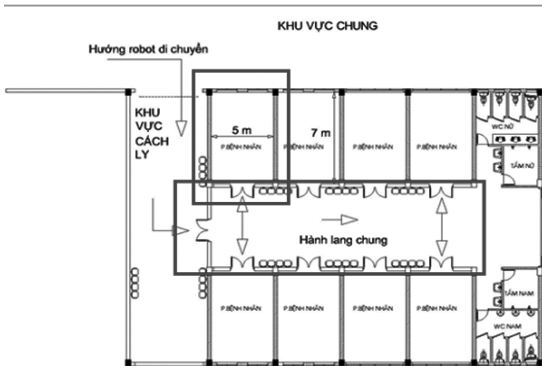
Hình 3.2. Kết quả bản đồ số trong mô phỏng

Sau đó, tiếp tục đặt vị trí đích và các vật cản xuất hiện ngẫu nhiên trong quá trình di chuyển, giả lập các thông tin cảm biến bằng công cụ Rviz. Kết quả cho thấy, robot đảm bảo di chuyển đến vị trí đích, tránh được các vật cản cố định và ngẫu nhiên. Quỹ đạo di chuyển như Hình 3.3.



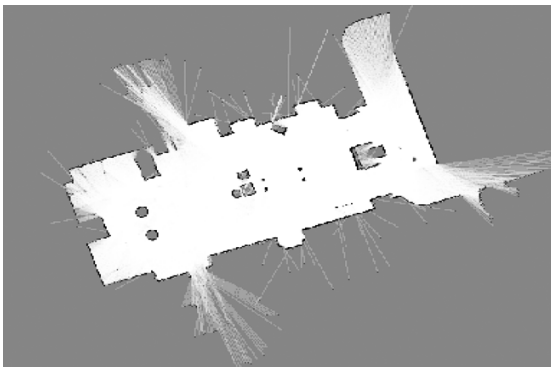
Hình 3.3. Quỹ đạo di chuyển trong mô phỏng

Từ đó, khẳng định tính đúng đắn của thuật toán, tiếp theo, nhóm tác giả đã tiến hành thử nghiệm thực tế tại Bệnh viện Bệnh Nhiệt đới Trung Ương – cơ sở Kim Chung – Đông Anh – Hà Nội. Trên Hình 3.4 là sơ đồ khu vực tiến hành thử nghiệm.



Hình 3.4. Khu vực thử nghiệm tại bệnh viện

Robot tiến hành quét môi trường phòng bệnh nhân và thu được kết quả bản đồ số như sau:



Hình 3.5. Kết quả bản đồ số của phòng bệnh

Và cuối cùng là xây dựng quỹ đạo lau sàn bất kỳ

(dựa trên việc thay đổi các điểm đích của quỹ đạo) cho robot và so sánh với quỹ đạo thực tế robot di chuyển.



Hình 3.6. Kết quả chạy của robot trong thực tế

Từ kết quả chạy thực tế, nhận thấy rằng về tổng thể, robot vẫn thực hiện đúng chu trình lau sàn được thiết lập, tuy nhiên, vẫn tồn tại một số vị trí lệch khỏi quỹ đạo. Điều này được giải thích bởi tấm lau nhà bên dưới robot bị chèn lên bánh xe và quán tính của lượng lớn dung dịch, khối lượng acquy khiến gây ra sự trượt khỏi quỹ đạo khi di chuyển. Các nguyên nhân này có thể khắc phục được thông qua chỉnh sửa cơ khí trong các thử nghiệm sau.

4. Kết luận

Bài báo đã trình bày về các kết quả trong việc chế tạo robot, đồng thời áp dụng các thuật toán và giải thuật vào trong thực tế và đã cho được kết quả khả quan. Tiếp theo, chúng tôi sẽ cố gắng hoàn thiện và tối ưu sản phẩm với mong muốn nghiên cứu này sẽ phục vụ tốt cho cộng đồng.

Lời cảm ơn

Nghiên cứu này được tài trợ bởi đề tài cấp bộ “Nghiên cứu, thiết kế chế tạo robot hỗ trợ nhân viên y tế trong việc khử khuẩn lau sàn nhà”.

Tài liệu tham khảo

- [1]. P. T. Cat, “Robotic research trends in the world and robotic research and development in Vietnam,” *Journal of Computer Science and Cybernetics*, vol. 3, pp. 197-212, 2010.
- [2]. L. a. B. S. Sciavicco, *Modelling and control of robot manipulators*, Springer Science & Business Media, 2012.
- [3]. A. C. K. Guilherme N. DeSouza, “Vision for Mobile Robot Navigation: A Survey,” *IEEE Transactions on Pattern and Machine Intelligence*, vol. 24, no. 2, pp. 237-267, 2002.
- [4]. A. S. Zimmerman, “Review of marine animals and bioinspired robotic vehicles: Classifications and characteristics,” *Progress in Aerospace Sciences*, vol. 93, pp. 95-119, 2017.
- [5]. E. G. C. L. Paolo Dario, “Humanoids and Personal Robots: Design and Experiments,” *Journal of Robotic Systems*, vol. 12, pp. 673–690, 2001.
- [6]. H. M. L. Ashutosh Singandhupe, “A Review of SLAM Techniques and Security in Autonomous Driving,” in *Third IEEE International Conference on Robotic Computing (IRC)*, 2019.
- [7]. S. a. M. L. Koenig, “D^{*} lite,” *Aaai/iaai*, vol. 15, 2002.
- [8]. V. H. T. T. H. N. Q. H. Nguyễn Quốc Hùng, “Đánh giá hiệu năng giải thuật Fab-map* định vị robot trong nhà sử dụng thông tin hình ảnh,” *Hội nghị Quốc gia lần thứ 9 về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR)*, Cần Thơ, 2016.
- [9]. S. T. D. K. B. W. Michael Montemerlo, “FastSLAM: A Factored Solution to the Simultaneous,” in *American Association for Artificial Intelligence*, 2002.
- [10]. S. Thrun, “Probabilistic Algorithms in Robotics,” *American Association for Artificial Intelligence*, vol. 21, no. 4, pp. 93-109, 2000.

**RESEARCH, DESIGN AND MANUFACTURE THE ROBOT-ASSISTED MEDICAL STAFF
IN DISINFECT AND MOP THE FLOOR**

Abstract:

The paper presents manufacturing the robot-assisted medical, combined with SLAM algorithms to locate and build digital maps, and the D Lite algorithm is used to calculate and optimize the motion trajectory of the robot. The robot has the ability to move automatically, avoid obstacles, choose the shortest path, automatically charge and perform medical assistance tasks. The robot's materials are suitable for the medical environment. The disinfectant spray system and mopping mechanism help ensure paramedic assistance in disinfecting and mopping the floor.*

Keywords: Robot-assisted medical, SLAM algorithm, D* Lite algorithm.