

# XÂY DỰNG QUỸ ĐẠO CỤC BỘ CHO ROBOT TỰ HÀNH TRONG NHÀ LƯỚI NÔNG NGHIỆP TRÊN NỀN TẢNG HỆ ĐIỀU HÀNH ROS

Phạm Thị Lan Hương\*, Nguyễn Thị Duyên

Khoa Cơ - Điện, Học viện Nông Nghiệp Việt Nam

\*Tác giả liên hệ: huongpham@vnua.edu.vn

Ngày nhận bài: 06.01.2023

Ngày chấp nhận đăng: 05.10.2023

## TÓM TẮT

Bài báo này trình bày về xây dựng quỹ đạo cục bộ nhằm tránh các vật cản tĩnh và động sử dụng thuật toán *TEB* (*Timed Elastic Bands*) dựa trên bản đồ, định vị đồng thời *SLAM* (*Simultaneous Localization and Mapping*) cho robot tự hành hoạt động trong các nhà lưới trồng cây nông nghiệp. Trên cơ sở robot Omni 4 bánh với nền tảng hệ điều hành lập trình cho robot ROS (*Robot Operating System*) và nền tảng máy tính nhúng hiệu suất cao *Jetson-TX2* xử lý tác vụ trung tâm, camera 3D và một cảm biến *Lidar* để thu thập dữ liệu từ môi trường nhà lưới nông nghiệp. Các kết quả chạy mô phỏng robot sử dụng phần mềm *Gazebo* và thử nghiệm trên phần mềm *Rviz* cho thấy sự hiệu quả của việc sử dụng hệ điều hành ROS, thuật toán *TEB* vào việc điều khiển, giám sát robot tự hành di chuyển linh hoạt có thể tránh được vật cản trong nhà lưới và là hướng nghiên cứu có nhiều tiềm năng trong các ứng dụng nông nghiệp công nghệ cao.

Từ khóa: Robot nông nghiệp, ROS, TEB, robot Omni...

## Constructing Local Orbit for Self-Operating Robot in Agricultural Greenhouse Based on ROS

## ABSTRACT

This paper proposed a local path planner coping with the collision-free navigation problem which considers both static and dynamic obstacles based on *Timed Elastic Bands (TEB)* aggregated with Simultaneously Localization and Mapping (*SLAM*) for self-operating/mobile robots working in the greenhouse environment. Moreover, the proposed system is a four-wheel Omni robot constructed using Robot Operating System (ROS) platform and implemented in a high-performance embedded computer *Jetson-TX2* connected with a 3D camera and lidar to collect environmental data. The simulation results based on *Gazebo* and *Rviz* software show the effectiveness of using the ROS operating system, with the *TEB* algorithm in controlling and monitoring autonomous moving robots, avoiding obstacles in the greenhouse This is a research direction with great potential in high-tech agricultural applications.

Keywords: Agricultural Robot, ROS, Timed Elastic Bands, Omni robot.

## 1. ĐẶT VẤN ĐỀ

Trong những năm gần đây, biến đổi khí hậu, gia tăng dân số và thiếu hụt lao động là những mối đe dọa cấp bách đến tính bền vững của nông nghiệp toàn cầu. Lo ngại về an ninh lương thực đã làm tăng sự quan tâm đến tự động hóa nông nghiệp, đến chi phí toàn cầu cho robot nông nghiệp đang tăng lên theo cấp số nhân (Bechar & Vigneault, 2016) và sự quan

tâm của các tổ chức trên toàn cầu đến việc phát triển tự động hóa nông nghiệp và robot nông nghiệp ngày càng tăng.

Canh tác kỹ thuật số là thực hiện các công nghệ hiện đại như cảm biến, robot và phân tích dữ liệu để chuyển từ các hoạt động cơ giới sang các quy trình tự động liên tục (Redmond, 2018; Bechar & Vigneault, 2016). Lĩnh vực robot nông nghiệp đã gặt hái được một số thành tựu, đặc biệt là về dò tìm, kiểm soát cỏ dại tự động và sâu

bệnh, phun thuốc có mục tiêu (Gonzalez-de-Soto & cs., 2016), cắt tỉa lá cây, xác định hình thái cây trồng (Zhang & cs., 2016) và thu hoạch. Xác định đối tượng, thuật toán lập kế hoạch nhiệm vụ, số hóa và tối ưu hóa cảm biến được nhấn mạnh là những thách thức phải đối mặt trong bối cảnh canh tác kỹ thuật số. Với những tiến bộ trong lý thuyết điều khiển, ứng dụng của những robot này trong canh tác kỹ thuật số ngày càng tăng mức độ tự động hóa, thay đổi các cách thức, công nghệ truyền thống sang ứng dụng công nghiệp công nghệ cao đang thu hút các nhà đầu tư, kỹ sư, nhà khoa học và doanh nghiệp. Các robot rất khó khăn để có thể tự động hóa hoàn toàn trong môi trường nông nghiệp. Một robot hoạt động trong môi trường nông nghiệp phải rất đa dạng, phi cấu trúc, với nhiều yếu tố bất định nhưng lại có các yêu cầu thao tác, cảm nhận và điều khiển yếu tố môi trường xung quanh một cách chính xác, điều này là thách thức lớn với các nhà nghiên cứu chế tạo (De Baerdemaeker & cs., 2001). Điển hình là ứng dụng robot trên cánh đồng khép kín như trong nhà lưới để canh tác cây rau và hoa quả trái vụ theo nghiên cứu của Shamshiri & cs. (2018).

Với robot nông nghiệp, việc ứng dụng các giải pháp cảm biến truyền thống là rất khó khả thi và xu hướng là robot được trang bị các loại cảm biến tiên tiến như camera 3D, laser scan để thu thập dữ liệu, bản đồ hóa và hoạt động tự động hoàn toàn. Durmus & cs. (2016) đã thực hiện dự án robot tự hành và thử nghiệm trong môi trường nhà lưới bằng cách sử dụng nền tảng ROS, robot di động sử dụng một camera 3D để lập bản đồ môi trường nhà lưới và điều hướng trong nhà lưới bằng cách sử dụng bản đồ đã tạo. Habibie & cs. (2018) đã sử dụng công cụ mô phỏng ROS để điều hướng một UGV trên cánh đồng cây ăn quả. Các đánh giá đều cho rằng robot hoạt động hiệu quả với độ chính xác cao trong quá trình canh tác.

Bên cạnh đó, trong nghiên cứu của Habibie & cs. (2018) và trong nghiên cứu của Park & Lee (2017) đã sử dụng phương pháp định vị và lập bản đồ đồng thời (Simultaneous Localization and Mapping - SLAM - là tạo ra bản đồ và phát hiện các chướng ngại vật xung quanh trong môi trường không xác định) là bước tiến trong định

vị, phát hiện các chướng ngại vật để tạo ra bản đồ. Bằng cách thu thập dữ liệu qua laser scan, IMU hoặc encoder thông qua ROS (Da Silva, 2019), vị trí của robot có thể được ước tính và tạo ra bản đồ môi trường đã tích hợp trong ROS. Từ các thông tin SLAM, cụ thể là bản đồ thu được, thiết lập các thuật toán xây dựng quỹ đạo tối ưu cho robot, hay là điều hướng cho robot, trên nền ROS có thể xây dựng các thuật toán điều hướng cho robot (Chen & cs., 2017; Priyandoko & cs., 2017; Zhi & Xuesong, 2018; Flavio & cs., 2018). Trong việc phát triển và điều khiển robot tự hành ứng dụng trong nhà lưới nông nghiệp, lập quỹ đạo chuyển động thông minh cho robot bao gồm xây dựng quỹ đạo toàn cục và quỹ đạo cục bộ, mà trong đó việc xác định quỹ đạo cục bộ cho robot, là cơ sở để đảm bảo robot có thể di chuyển tới vị trí đích đồng thời để hoàn thiện bài toán điều hướng robot. Từ những phân tích từ trên, bài báo tập trung trình bày về xây dựng quỹ đạo cục bộ cho robot tự hành trong nhà lưới nông nghiệp trên nền tảng hệ điều hành ROS.

## 2. PHƯƠNG PHÁP NGHIÊN CỨU

Tác giả đã tiến hành tham khảo các tài liệu, công trình nghiên cứu về thuật toán điều khiển robot đã xuất bản trong và ngoài nước, đồng thời kế thừa các kết quả nghiên cứu đã có để tiến hành nghiên cứu, lựa chọn thuật toán phù hợp để nghiên cứu hệ thống định vị, dẫn đường, lập bản đồ, tránh chướng ngại vật của robot tự hành 4 bánh hoàn chỉnh di chuyển trong nhà lưới. Cuối cùng, sử dụng máy tính có hệ điều hành ROS, phần mềm mô phỏng Rviz (3D Visualization Tool); phần mềm mô phỏng Gazebo để tiến hành mô phỏng xây dựng quỹ đạo cục bộ cho robot tự hành trong nhà lưới.

Hệ điều hành Robot ROS (Robot Operating System) là một mã nguồn mở, hệ điều hành cho robot cung cấp các dịch vụ, bao gồm kết nối tới phần cứng, điều khiển thiết bị cấp thấp và thực hiện những chức năng mà robot thường sử dụng. Nó cũng cung cấp các công cụ và thư viện để lấy, xây dựng, viết và chạy mã trên nhiều máy tính. ROS tương tự như hệ điều hành. Tuy nhiên, không giống như các hệ điều hành thông

thường, nó có thể được sử dụng nhiều kết hợp tác vụ cho phần cứng. Hơn nữa, nó là một phần mềm Robot nền tảng cung cấp các môi trường phát triển khác nhau chuyên phát triển các chương trình cho Robot ứng dụng.

Phần mềm mô phỏng Rviz (3D Visualization Tool) là công cụ trực quan của ROS, hiển thị các thông báo thu được ở chế độ 3D, cho phép người dùng xác minh trực quan dữ liệu. Rviz có thể hình dung khoảng cách từ cảm biến Laser Distance Sensor (LDS) đến vị trí của vật cản, Point Cloud Data (PCD) của cảm biến khoảng cách 3D như RealSense, Kinect hoặc Xtion, giá trị hình ảnh thu được từ camera hoặc có thể nhiều hơn mà không cần sử dụng đến phần mềm khác.

Phần mềm mô phỏng Gazebo là một công cụ hỗ trợ không thể thiếu được trong quá trình thiết kế robot. Gazebo là một trình giả lập 3D cung cấp robot, cảm biến, mô hình môi trường để mô phỏng 3D cần thiết cho sự phát triển của robot và cung cấp mô phỏng thực tế với động cơ vật lý của nó. Gazebo là một trong những trình giả lập phổ biến nhất có hiệu suất cao, mã nguồn mở trong những năm gần đây và đã được chọn làm trình giả lập chính thức của DARPA Robotics Challenge<sup>7</sup> tại Mỹ. Hơn thế nữa, Gazebo được phát triển và phân phối bởi Open Robotics, phụ trách bởi cộng đồng ROS, vì vậy nó có tính tương thích cao với ROS.

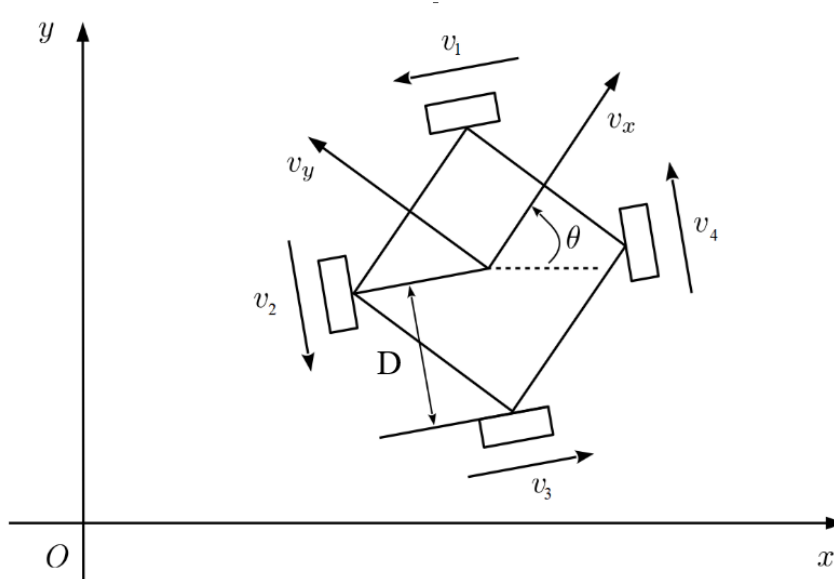
### 3. KẾT QUẢ VÀ THẢO LUẬN

#### 3.1. Mô hình robot tự hành Omni bốn bánh

Omni Robot tự hành đa hướng có khả năng di chuyển theo bất kỳ hướng nào mà không cần phải thay đổi vị trí và tư thế. Với cấu trúc bánh khác biệt và ưu điểm về khả năng di chuyển vượt trội trong các điều kiện môi trường di chuyển hẹp, khó thay đổi vị trí, mẫu mobile robot này đang được áp dụng một cách rộng rãi không chỉ trong nghiên cứu mà còn trong các lĩnh vực sản xuất và đời sống. Đây chính là lý do chính mà bài báo muốn đề cập đến. Mô hình động học cho Robot Omni bốn bánh được xây dựng dựa trên mô hình bánh xe Omni được bố trí đặt cách nhau một góc 90° như trong hình 1.

$q = [x \ y \ \theta]^T$  là vectơ tọa độ của robot trong hệ tọa độ toàn cục Oxy, trong đó x và y lần lượt là tọa độ của robot theo phương Ox và Oy,  $\theta$  là góc quay của robot so với chiều dương của trục Ox.  $v = [v_x \ v_y \ \omega]^T$  là vectơ vận tốc của robot trong hệ trục tọa độ động gắn vào tâm robot bao gồm vận tốc tịnh tiến và vận tốc góc của robot.

Để đơn giản trong việc thiết kế bộ điều khiển robot, mối quan hệ giữa vận tốc trong trục của robot và vận tốc trong trục của môi trường được mô tả bằng mô hình động học của robot:



Hình 1. Robot Omni đa hướng 4 bánh

$$\dot{\underline{q}} = H \cdot \underline{v} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \underline{v} \quad (1)$$

trong đó:  $H = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$  là ma trận

chuyển hệ trục tọa độ.

Từ (1) chúng ta sẽ có được phương trình để lập trình cho robot trong ROS:

$$\begin{cases} \dot{x} = v_x \cos\theta - v_y \sin\theta \\ \dot{y} = v_x \sin\theta + v_y \cos\theta \\ \dot{\theta} = w \end{cases} \quad (2)$$

Sau khi tín hiệu được chuyển đổi, vị trí của robot được điều khiển trực tiếp bởi các tín hiệu này thông qua bốn bánh của robot.

Mô hình robot được xây dựng với 3 thành phần chính: phần xử lý trung tâm, phần cảm biến và phần điều khiển cơ cấu chấp hành. Cấu trúc bộ phận di chuyển là Omni 4 bánh đa hướng giúp việc di chuyển linh hoạt và dễ dàng hơn. Máy tính nhúng đóng Jetson TX2 có vai trò là bộ xử lý trung tâm được cài đặt hệ điều hành ROS cùng với các node tính toán thuật toán SLAM. Sau khi tính toán xong, Jetson TX2 gửi lệnh điều khiển cho bộ phận điều khiển cơ cấu chấp hành là mạch STM32. RPLidar quét laser 360° giúp robot xây dựng bản đồ và nhận diện vật cản tầm cao với khoảng cách xa. Camera 3D sử dụng Deep Camera giúp nhận diện vật cản tầm trung và tầm thấp ở phía trước robot.

### 3.2. Bản đồ hóa môi trường và định vị đồng thời SLAM cho robot

Trong lĩnh vực robotics, vấn đề định vị đồng thời xây dựng bản đồ SLAM là một trong những vấn đề quan trọng nhất và đóng vai trò then chốt trong điều hướng robot. Vì vậy đây là vấn đề đã thu hút được sự quan tâm lớn của các nhà khoa học. Vấn đề SLAM được mô tả tổng quát trong quá trình khi robot di chuyển để lập bản đồ những nơi con người không thể hoặc không muốn tiếp cận, đồng thời robot tự xác định được vị trí của nó so với những đối tượng xung quanh. Kỹ thuật SLAM cung cấp thông tin bản đồ về môi trường đồng thời sử dụng bản đồ này ước tính tư thế riêng (vị trí và hướng) của robot dựa vào các tín hiệu thu được từ các cảm biến tầm nhìn bao gồm Rplidar và 3D camera.

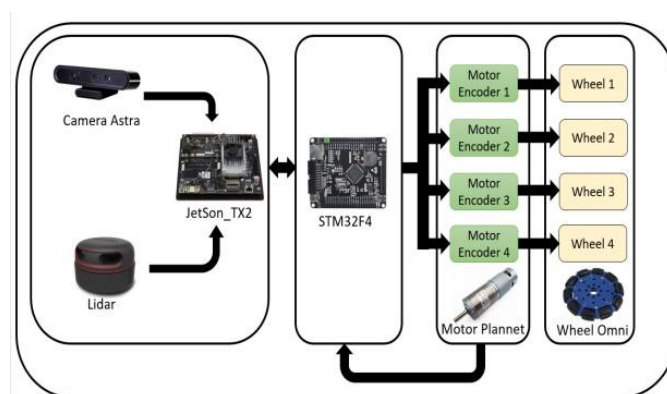
Để thực hiện SLAM, chúng tôi sử dụng gói Gmapping cho omni robot. Bằng việc thu các dữ liệu từ RPLidar, Astra camera và chuyển Deep to Laser với cấu trúc các phần như hình 3.

Gói Gmapping ước tính vị trí của robot và xây dựng bản đồ dựa trên dữ liệu thu được và các phép đo hình học của nó.

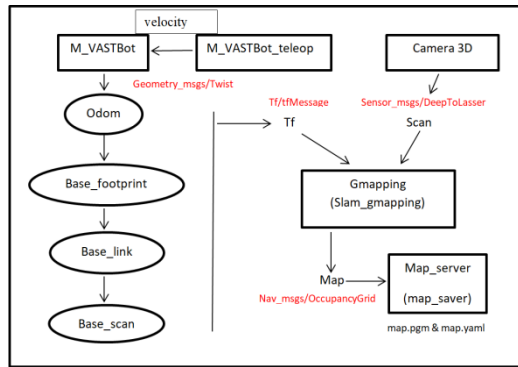
- Thư viện TF được thiết kế và cung cấp các tiêu chuẩn để theo dõi các khung tọa độ và chuyển đổi dữ liệu trong toàn bộ hệ thống.

- RPLidar: node này thực hiện chạy cảm biến RPLidar và gửi thông tin “scan” cần thiết cho SLAM đến node Gmapping.

- Teleop: node này là thuật toán điều khiển để robot có thể di chuyển theo ý muốn của người dùng. Sau đó tiến hành gửi vận tốc góc và vận tốc di chuyển cho core dựa trên tín hiệu thu được.



Hình 2. Mô hình thực tế và cấu trúc phần cứng của Omni robot



Hình 3. Sơ đồ tín hiệu xây dựng bản đồ, định vị và điều hướng robot

- Core: node này nhận được vận tốc di chuyển và vận tốc góc. Trong khi Publishes “odom”, đây là tư thế được đo và ước tính của robot. Bên cạnh đó còn publishes tọa độ của robot đã được chuyển đổi theo thứ tự:

Odom → Base\_footprint → Base\_link → Base\_scan. Sau đó những dữ liệu này sẽ được gửi đến topIC tf.

- Gmapping: node này sẽ tạo ra bản đồ dựa trên thông tin đo đạc khoảng cách từ tín hiệu laser scan và thông tin từ topIC tf, đó là tư thế của robot.

- Map\_server: node này tiến hành tạo file “map.gpm” và file “map.yaml”, 2 file này chứa các thông tin của bản đồ thu được.

### 3.3. Lập quỹ đạo cục bộ cho robot tự hành trên nền tảng hệ điều hành ROS

Khi robot đã có thông tin về môi trường và vị trí của nó từ hệ thống SLAM, các vị trí đích có thể được đặt nhưng chỉ trong vùng môi trường đã xác định. Cần thiết lập quỹ đạo toàn cầu (Global planner) và thiết lập quỹ đạo cục bộ (Local planner) cho hệ thống dẫn hướng điều khiển (như thể hiện trong Hình 4) với mục tiêu đặt ra robot có thể di chuyển tự động (theo quỹ đạo từ vị trí xuất phát đến vị trí mục tiêu) và an toàn (tránh được những vật cản tĩnh và vật cản động của môi trường thực). Trước tiên, robot cần xác định được vị trí của nó trong bản đồ tại thời điểm hiện tại. Khi đó robot sẽ có những nhận thức về môi trường xung quanh, đặc biệt là bản đồ bao gồm những vật cản tĩnh đã xác định được

ở hệ thống SLAM. Tiếp theo, vị trí đích được xác định và đặt trong vùng phạm vi đã xác định của bản đồ. Từ đó, các thuật toán tính toán quỹ đạo sẽ tính toán để đảm bảo robot có thể di chuyển tới vị trí đích với quỹ đạo ngắn nhất.

Phần lập quỹ đạo toàn cục được thiết kế sử dụng thuật toán A\* (Chen & cs., 2017) dựa trên nền tảng ROS. Thuật toán A\* xây dựng tăng dần tất cả các tuyến đường từ điểm xuất phát cho tới khi nó tìm thấy một đường đi chạm tới đích. Thuật toán này sử dụng một đánh giá “heuristic” về khoảng cách từ điểm bất kỳ cho trước tới đích để biết được những đường nào có khả năng dẫn tới đích, bên cạnh đó thuật toán này còn tính đến khoảng cách nó đã đi qua.

Trong bài báo này, chúng tôi sẽ đi sâu vào đề cập đến bài toán lập kế hoạch đường đi cục bộ (Local planner) sử dụng thuật toán TEB (Timed Elastic Bands) (Rösmann, 2017).

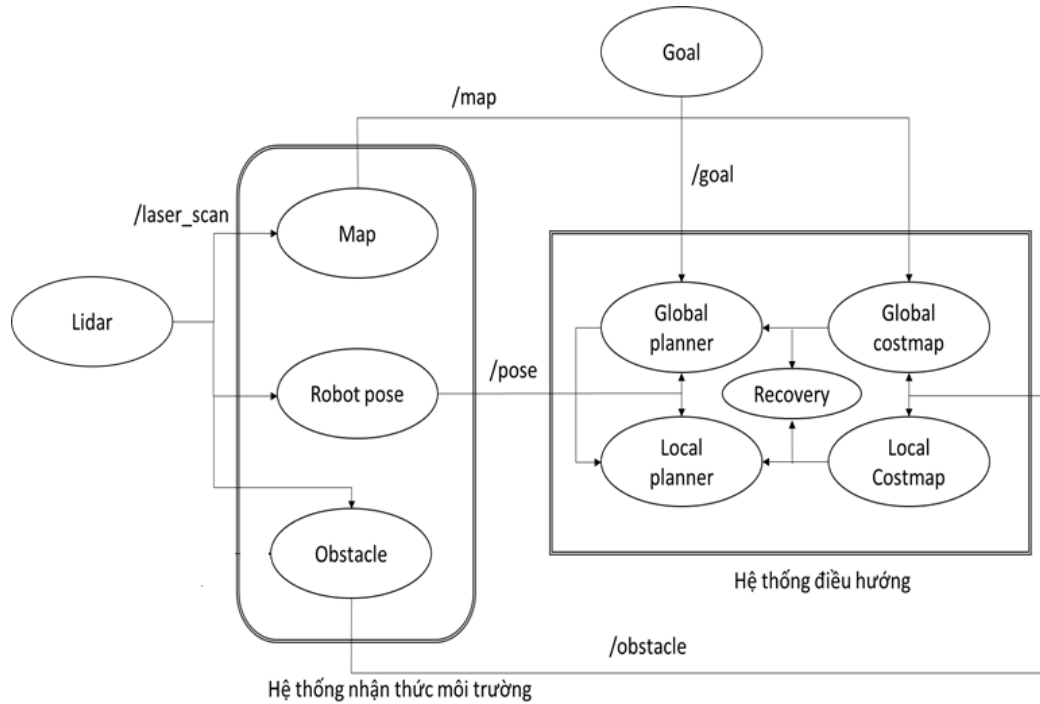
Hình 5 chỉ ra vị trí của robot trong các thời điểm khác nhau trên bản đồ. Ta định nghĩa  $s_i = [x_i \ y_i \ \beta_i] \in \mathbb{R}^2 \times S^1$  với  $i = 1, 2, \dots, N$  là tọa độ của robot tại thời điểm  $i$  trong bản đồ toàn cục, trong đó  $x_i, y_i$  là tọa độ của robot theo 2 phương  $x, y$  và  $\beta_i$  là hướng của robot (là góc xác định bởi đường thẳng tạo bởi tọa độ  $(x_i, y_i)$  với vị trí đích với phương  $Ox$ ).

$$Q\{S_i\}_{i=0, \dots, 1} \quad n \in N \quad (2)$$

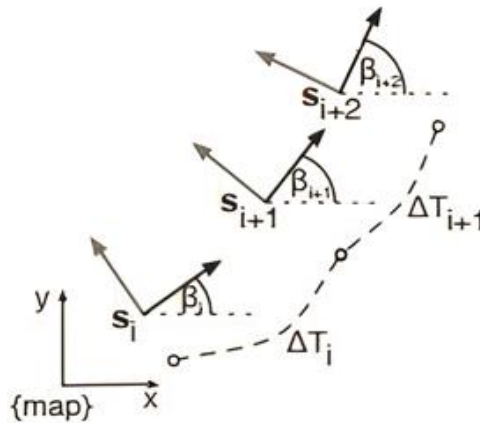
$$\tau = \{\Delta T_1\}_{i=0, \dots, 1-n} \quad (3)$$

Thuật toán TEB được định nghĩa là:

$$B := (Q, \tau) \quad (4)$$



Hình 4. Sơ đồ hệ thống điều hướng cho robot



Hình 5. Vị trí của robot tại các thời điểm khác nhau trên bản đồ

Hàm mục tiêu:

$$f(B) = \sum_k \beta_k f_k(B) \quad (5)$$

$$B^* = \arg \min_B f(B) \quad (6)$$

Trong đó  $f(B)$  là hàm mục tiêu toàn cục,  $B^*$  là giá trị tối ưu của thuật toán TEB.

Mục tiêu của thuật toán TEB là tìm ra giá trị  $B^*$  để sinh ra đường đi tối ưu cho robot. Các hàm mục tiêu  $f_k(B)$  được tính toán nhằm giới hạn được vận tốc và gia tốc của robot qua đó tối

ưu hóa được quãng đường di chuyển đến đích của robot với quãng đường đi ngắn nhất và trong thời gian ngắn nhất, đồng thời không va chạm với các vật cản trong môi trường.

Ta định nghĩa hàm mục tiêu  $f_k(B)$  dựa trên hàm mất mát  $e_T$ :

$$e_T(x, x_r, \varepsilon, S, n) = \begin{cases} \left( \frac{x - (x_r - \varepsilon)}{S} \right)^n & ; \quad x > x_r - \varepsilon \\ 0 & \end{cases} \quad (7)$$

Trong đó  $x_r$  là giá trị giới hạn,  $S$ ,  $n$  thể hiện độ chia và bậc của đa thức,  $\varepsilon$  là số dương nhỏ thể hiện sai lệch xấp xỉ.

Các hàm mục tiêu được lựa chọn và tính toán dựa trên trạng thái và vị trí hiện tại của robot trên bản đồ toàn cục.

Hình 6 minh họa các điểm đặt mong muốn và vật cản trên quãng đường di chuyển của robot, dựa vào việc tối ưu các hàm mục tiêu thuật toán TEB sinh ra đường đi tốt nhất cho robot bám theo quỹ đạo mong muốn đồng thời tránh các vật cản trên đường đi. Để tìm ra được các điểm đặt mong muốn (WP) và tránh được các vật cản, thuật toán TEB sử dụng hai hàm mục tiêu là:

$$f_{\text{path}} = e_{\Gamma}(d_{\text{min}}, r_{\text{wpmax}}, \varepsilon, S, n) \quad (8)$$

$$f_{\text{ob}} = e_{\Gamma}(-d_{\text{min}}, -r_{\text{omin}}, \varepsilon, S, n) \quad (9)$$

Trong đó  $d_{\text{min}}$  là khoảng cách nhỏ nhất từ vị trí của robot đến điểm đặt hoặc vật cản,  $r_{\text{wpmax}}$  là giới hạn khoảng cách lớn nhất của robot so với điểm đặt,  $r_{\text{omin}}$  là giới hạn khoảng cách an toàn của robot đến vật cản.

Thuật toán TEB sinh ra quỹ đạo tránh các vật cản động cũng như vật cản tĩnh trong quá trình di chuyển của robot đồng thời tính ra quỹ đạo ngắn nhất cho robot di chuyển đến đích. Ta có  $(\Delta T_k)_{k=1,2,\dots,n-1}$  là tập hợp thời gian robot di chuyển hết quãng đường, trong đó  $\Delta T_k$  là khoảng thời gian robot di chuyển từ vị trí  $s_k$  đến vị trí  $s_{k+1}$ . Thuật toán TEB tính ra đường đi nhanh nhất cho robot thông qua việc tính toán tối ưu cực tiểu  $\min_B(f_{\text{time}})$  với:

$$f_{\text{time}} = \left( \sum_{i=1}^n \Delta T_i \right)^2 \quad (10)$$

Việc tính toán vận tốc dài và vận tốc góc của robot dựa vào khoảng cách giữa hai vị trí  $s_k$  và  $s_{k+1}$  trong khoảng thời gian di chuyển là  $\Delta T_k$

- Giới hạn vận tốc và gia tốc của robot:

Việc tính toán vận tốc dài và vận tốc góc của robot dựa vào khoảng cách giữa hai vị trí  $s_k$  và  $s_{k+1}$  trong khoảng thời gian di chuyển là  $\Delta T_k$

$$v_i = \frac{1}{\Delta T_i} \left\| \begin{pmatrix} x_{i+1} - x_i \\ y_{i+1} - y_i \end{pmatrix} \right\| \quad (11)$$

$$\omega_i = \frac{\beta_{i+1} - \beta_i}{\Delta T_i} \quad (12)$$

Ta có  $|\omega_i| \leq \omega_{\text{max}}$  với  $\omega_{\text{max}} = \frac{v_{\text{max}}}{d_{\text{min}}}$ . Giới hạn  $\pm v_{\text{max}}$  và  $\pm \omega_{\text{max}}$  được áp dụng để giới hạn

$$v_k(s_{k+1}, s_k, \Delta T_k) = \left[ v_{\text{max}} - |v_k|, \omega_{\text{max}} - |\omega_k| \right]^T.$$

Tương tự ta có gia tốc của robot dạng:

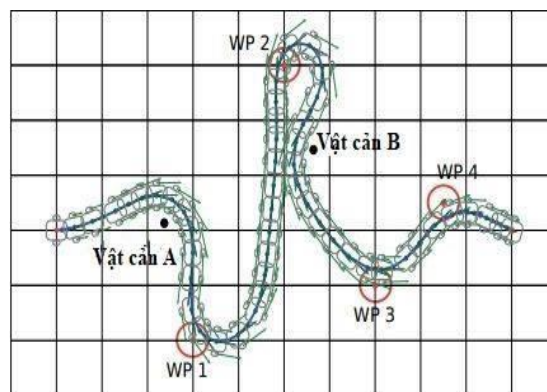
$$a_i = \frac{2(v_{i+1} - v_i)}{\Delta T_i + \Delta T_{i+1}} \quad (13)$$

Tương tự ta tính được gia tốc góc của robot bằng việc thay thế vận tốc dài  $v_i$  bằng vận tốc góc  $\omega_i$ . Ta có mối quan hệ giữa vận tốc dài  $v_i$  với vận tốc góc  $\omega_i$  của các bánh xe.

$$v_{w,i} = v_i + L\omega_i \quad (14)$$

$$v_{w,i} = v_i - L\omega_i \quad (15)$$

trong đó  $L$  là bán kính bánh xe.



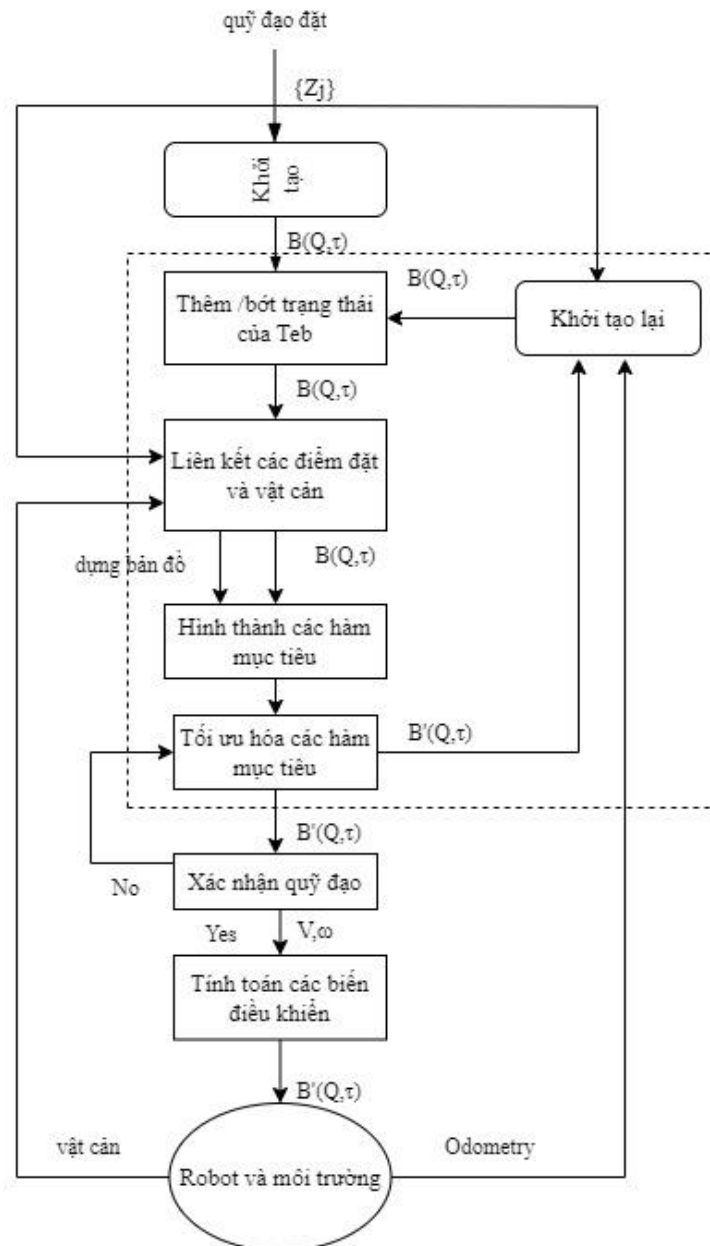
Hình 6. Thuật toán TEB

- Áp dụng thuật toán TEB cho robot Omni

Hình 7 cho thấy cấu trúc điều khiển cho robot Omni sử dụng thuật toán TEB.

Tại bước khởi tạo, thuật toán TEB điều chỉnh độ phân giải về không gian và thời gian cho quỹ đạo của robot thông qua các trạng thái của robot, sau đó liên kết các điểm đặt trên quỹ đạo đường đi của robot và vật cản trong môi trường với từng trạng thái của thuật toán TEB. Dựa vào việc nhận thức về môi trường và đường đi của robot, các hàm mục tiêu được hình thành

và liên kết với nhau, việc tối ưu các hàm mục tiêu sẽ hình thành nên quỹ đạo đường đi tốt nhất cho robot đồng thời robot có thể tránh các vật cản tĩnh cũng như vật cản động trong suốt quá trình di chuyển. Dựa vào mô hình động học của robot, các trạng thái của robot tại mỗi thời điểm sẽ chuyển thành các tín hiệu điều khiển về vận tốc và gia tốc cho robot. Sau mỗi vòng lặp, vị trí mới của robot và vật cản trong bản đồ toàn cục được cập nhật lại để thuật toán hình thành đường đi tiếp theo cho robot.



Hình 7. Cấu trúc điều khiển cho robot sử dụng TEB



### 3.4. Kết quả mô phỏng và thực nghiệm

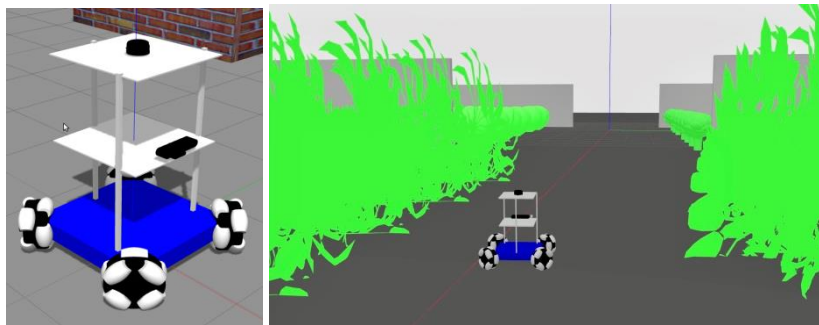
Phần mềm mô phỏng Gazebo được tích hợp để có thể sử dụng trong ROS. Môi trường trong Gazebo được tối ưu sao cho các điều kiện vật lý giống với môi trường thực tế nhất, nhóm tác giả đã xây dựng môi trường tại nhà lưới nhằm đa dạng các môi trường hoạt động trong nhà lưới với những loại cây trồng khác nhau. Mô hình robot tự hành Omni như hình 8, với các tham số mô phỏng và thực nghiệm giống nhau:

- Tốc độ lớn nhất theo phương x và y: 1,5 m/s
- Tốc độ góc lớn nhất: 0,5 rad/s
- Bán kính thân robot (D - Hình 1): 0,25m
- Bán kính bánh xe: 0,07m
- Tham số cho cảm biến lidar: Phạm vi quét lớn nhất: 0,2 ÷ 10m, độ phân giải: 1°, góc quét: 360°

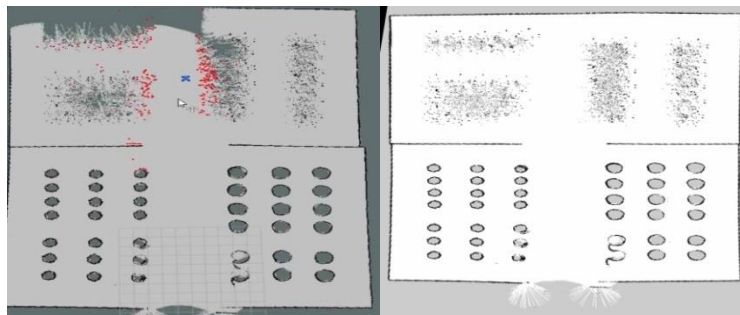
Việc chạy mô phỏng thử nghiệm SLAM được thực hiện trên Rviz, là công cụ trực quan của ROS. Mục đích chính là hiển thị các thông báo thu được ở chế độ 3D, cho phép người dùng xác minh trực quan dữ liệu. Thông qua Rviz, người dùng có thể giám sát được môi trường xung quanh robot khám phá được theo thời gian. Hình 9 là kết quả thu được khi bắt khởi chạy

kỹ thuật SLAM trong hệ thống nhận robot. Các vùng màu trắng là các vùng môi trường đã được nhận thức. Các đường màu đen là các vật cản (tường, các cây trồng) mà robot nhận dạng được và bản đồ hoá.

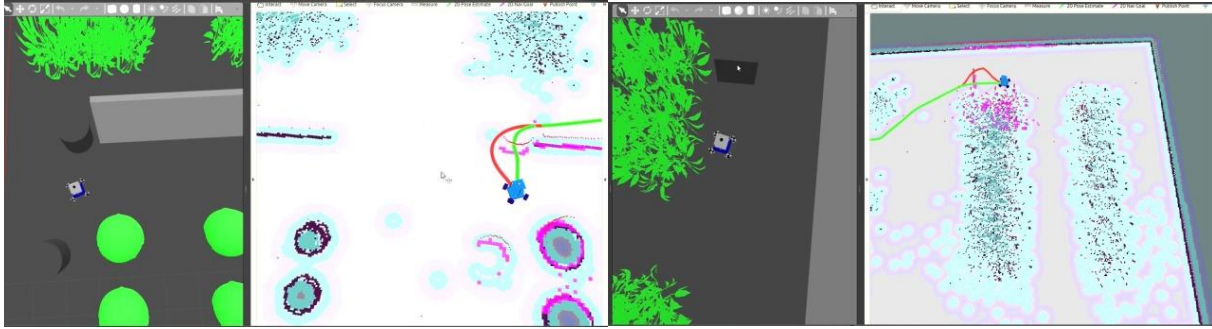
Hình 10 cho thấy hệ thống điều hướng hoàn toàn tự động, quy hoạch đường đi và xác định tuyến đường cho robot Omni di chuyển theo để đạt được vị trí mong muốn. Hệ thống điều hướng sẽ tính toán hai loại quỹ đạo. Quỹ đạo toàn cục cho robot (đường màu xanh) là quỹ đạo được tính toán sao cho nó là quỹ đạo ngắn nhất từ vị trí hiện tại của robot tới đích. Quỹ đạo cục bộ (đường màu đỏ) là quỹ đạo đặt cho việc điều khiển hệ thống robot. Quỹ đạo này sẽ điều chỉnh quỹ đạo của robot sao cho robot có thể di chuyển an toàn, tránh va chạm với vật cản mà vẫn đảm bảo robot tiến tới vị trí đích. Khi global planner đã được tạo ra thì local planner sẽ chuyển con đường này thành các lệnh vận tốc cho động cơ robot. Khi robot đi tới một góc tường, nhận thấy có khả năng robot sẽ va chạm với vật cản này, phần lập quỹ đạo cục bộ ngay lập tức điều chỉnh lại quỹ đạo cho robot. Quỹ đạo này khiến robot di chuyển vòng xa góc tường hơn nhưng vẫn đảm bảo bám sát tới quỹ đạo toàn cục.



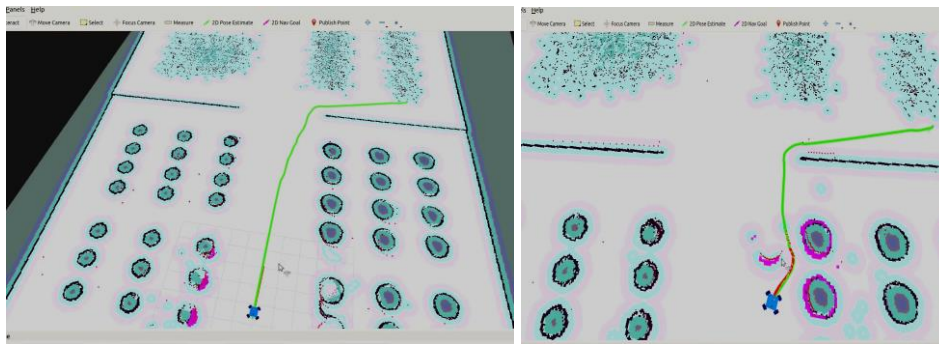
Hình 8. Mô hình 3D robot Omni và môi trường thực nghiệm tại mô hình nhà lưới



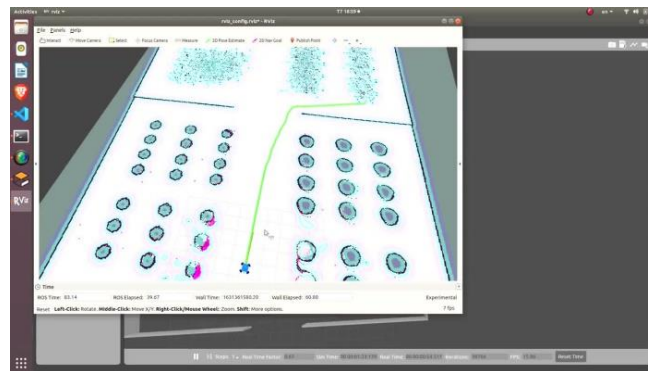
Hình 9. Quá trình SLAM của robot và kết quả bản đồ hoá môi trường tại nhà lưới



**Hình 10. Robot di chuyển tới vị trí đích và quỹ đạo cục bộ tránh các vật cản trên đường đi trên Gazebo và Rviz**



**Hình 11. Robot di chuyển tới vị trí đích trên Rviz**



**Hình 12. Quỹ đạo dự kiến robot sẽ đi**

Khi hoạt động, các cảm biến cập nhật bản đồ với các vật cản mới phát hiện được. Costmap sử dụng thông tin từ cảm biến tạo ra bản đồ cục bộ xung quanh robot Omni. Khối lập quỹ đạo cục bộ (Local planner) căn cứ vào vật cản gặp phải và thuật toán TEB sẽ cho ra kế hoạch di chuyển với khoảng cách có thể thay đổi, ở đây khoảng cách được chọn là 0,5m. Như trong Hình 11 thì sau khi tránh vật cản thì robot sẽ có thể lại di chuyển theo con đường mong muốn trong thời gian ngắn và đi tới mục tiêu.

Một số hình ảnh robot khi chạy mô phỏng tránh vật cản trong nhà lưới thông qua các kịch bản đặt ra được trình bày qua các hình 12 đến hình 16.

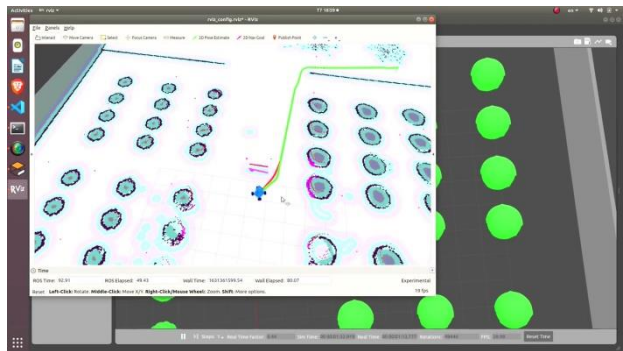
Quỹ đạo dự kiến robot sẽ di chuyển trong nhà lưới (Hình 12).

Trong quá trình di chuyển, robot gặp vũng nước (Hình 13, 14).

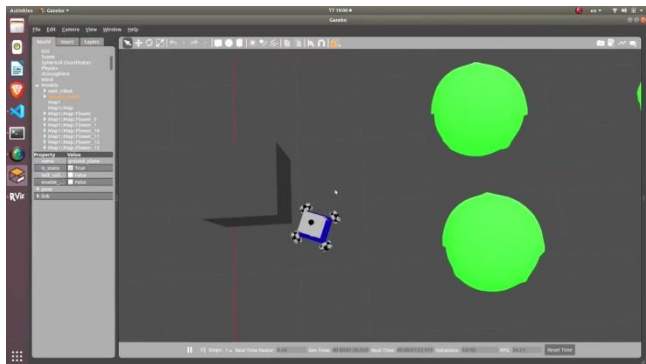
Khi gặp mô đất cao (Hình 15).

Quá trình điều chỉnh khi gặp mô đất cao (Hình 16).

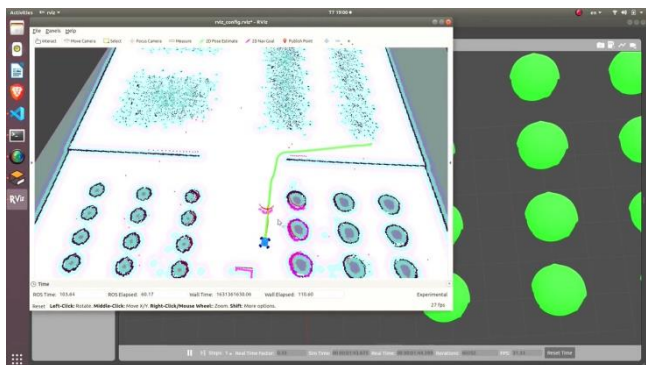
Xây dựng quỹ đạo cục bộ cho robot tự hành trong nhà lưới nông nghiệp trên nền tảng hệ điều hành ROS



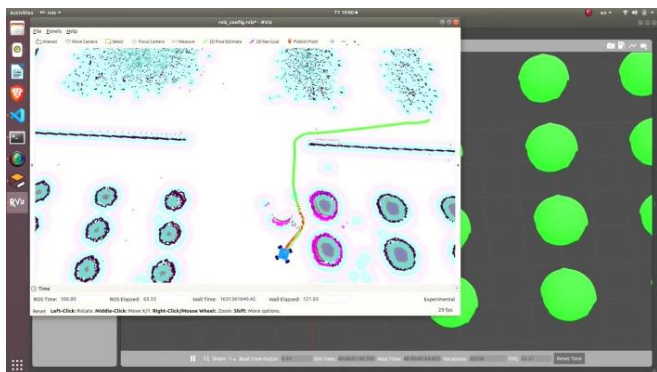
**Hình 13. Quỹ đạo của robot khi gặp vũng nước**



**Hình 14. Robot điều chỉnh quỹ đạo tránh vũng nước**



**Hình 15. Robot gặp mô đất cao**



**Hình 16. Quỹ đạo của robot khi gặp mô đất cao**

#### 4. KẾT LUẬN

Các kết quả cho thấy hiệu quả của thuật toán TEB trong việc lập quỹ đạo chuyển động cục bộ dựa trên trên bản đồ 2D thu được từ quá trình SLAM sử dụng hàm Gmapping. Các kết quả này là nền tảng cho các bước điều hướng, lập quỹ đạo chuyển động thông minh cho robot phục vụ các bài toán cụ thể trong nhà lưới, như ứng dụng trong công việc chăm sóc cây trồng (giám sát quá trình sinh trưởng, tưới nước, cung cấp dinh dưỡng hoặc phun thuốc sâu) và thu hoạch trong nhà lưới. Kết quả của nghiên cứu sẽ được ứng dụng cụ thể trong việc chăm sóc cây hoa lan được trồng trong nhà lưới tại Học viện Nông nghiệp Việt Nam.

#### TÀI LIỆU THAM KHẢO

- Bechar A. & Vigneault C. (2016). Agricultural robots for field operations: Concepts and components. *Biosystems Engineering*. 149: 94-111.
- Chen Y., Wang X., Hong S., Zhong X. & Zou C. (2017). Motion planning implemented in ROS for mobile robot. 29th Chinese Control And Decision Conference (CCDC), Chongqing. pp. 7149-7154.
- Da Silva B.M., Xavier R.S. & Gonçalves L.M. (2019). Mapping and Navigation for Indoor Robots under ROS. An Experimental Analysis, Creative Commons CC BY license.
- De Baerdemaeker J., Munack A., Ramon H. & Speckmann H. (2001). Mechatronic systems, communication, and control in precision agriculture. *IEEE Control Syst*. 21(5): 48-70.
- Durmus H., Gunes E.O. & Kirci M. (2016). Data acquisition from greenhouses by using autonomous mobile robot. In Proceedings of the 5<sup>th</sup> International Conference on Agro-Geoinformatics, Tianjin, China, 18-20 July 2016. pp. 5-9.
- Flavio B.P. Malavazi, Remy Guyonneau, Jean-Baptiste Fasquel, Sebastien Lagrange & Franck Mercier (2018). LiDAR-only based navigation algorithm for an autonomous agricultural robot. *Computers and Electronics in Agriculture*. 154: 71-79.
- Gonzalez-de-Soto M., Emmi L., Perez-Ruiz M., Aguera J. & Gonzalez-de-Santos P. (2016). Autonomous systems for precise spraying - Evaluation of a robotised patch sprayer. *Biosyst, Eng*. 146: 165-182.
- Habibie N., Nugraha A.M., Anshori A.Z., Masum M.A. & Jatmiko W. (2018). Fruit mapping mobile robot on simulated agricultural area in Gazebo simulator using simultaneous localization and mapping (SLAM). In Proceedings of the 2017 International Symposium on Micro-NanoMechatronics and Human Science (MHS), Nagoya, Japan, -6 December. pp. 1-7.
- Park S. & Lee G. (2017). Mapping and localization of cooperative robots by ROS and SLAM in unknown working area. In 2017 56<sup>th</sup> Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE). pp. 858-861: IEEE.
- Priyandoko G., Ming T. & Achmad M. (2017). Mapping of unknown industrial plant using ROS-based navigation mobile robot. In IOP Conference Series: Materials Science and Engineering. 257(1): 012088. IOP Publishing.
- Rösmann C.F.H.a.T.B. (2017). Online trajectory planning in ROS under kinodynamic constraints with timed-elasticbands. *Robot Operating System (ROS)*. Springer, Cham. pp. 231-261.
- Shamshiri R.R., Weltzien C., Hameed I.A., Yule I.J., Grift T.E., Balasundram S.K., Pitonakova L., Ahmad D. & Chowdhary G. (2018). Research and development in agricultural robotics: A perspective of digital farming. *Int J Agric & Biol Eng*. 11(41293): 1-11.
- Shamshiri R.R., Kalantari F., Ting K.C., Thorp K.R., Hameed I.A., Weltzien C., Ahmad D. & Shad Z.M. (2018). Advances in greenhouse automation and controlled environment agriculture: A transition to plant factories and urban agriculture. *Int J Agric & Biol Eng*. 11(1): 1-22.
- Zhang C., Gao H., Zhou J., Cousins A., Pumphrey M.O. & Sankaran S. (2016). 3D robotic system development for high-throughput crop phenotyping. *IFAC-PapersOnLine*. 49(16): 242-247.
- Zhi L. & Xuesong M. (2018). Navigation and Control System of Mobile Robot Based on ROS in 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). pp. 368-372: IEEE.