

# TÌM ĐƯỜNG ĐI CHO ROBOT DI ĐỘNG SỬ DỤNG KỸ THUẬT RRT VÀ PSO FINDING THE PATH FOR MOBILE ROBOT USING RRT AND PSO TECHNIQUE

Tổng Thành Trung  
Ngô Văn Thuyền

## TÓM TẮT

Cây RRT (*Rapidly-exploring Random Tree*) là một cấu trúc dữ liệu ngẫu nhiên và hiệu quả thiết kế dành cho việc tìm kiếm trong không gian đa chiều. RRT có thể được coi như là một kỹ thuật để tạo ra những điểm ngẫu nhiên trong hệ thống phi tuyến, đặc biệt thích hợp cho vấn đề lập quỹ đạo đường đi trong không gian có vật cản và các ràng buộc nonholonomic và kinodynamic. Thuật toán PSO (*Particle Swarm Optimization*) là thuật toán hữu hiệu cho bài toán tìm kiếm tối ưu.

Bài báo này trình bày việc sử dụng cấu trúc cây RRT kết hợp với thuật toán PSO để tìm đường đi cho robot trong môi trường cho trước. Thuật toán PSO được sử dụng để tìm vận tốc gốc và vận tốc thẳng cho robot để robot có thể di chuyển từ điểm ban đầu đến điểm kế tiếp trên cây. Kết quả mô phỏng trên Matlab cho thấy tính hiệu quả của phương pháp kết hợp cây RRT và PSO tìm đường đi cho robot di động.

## ABSTRACT

RRT (*Rapidly-exploring Random Tree*) is a random data structure, designed for efficient search in the multi-dimensional space. RRT can be regarded as a technique to create random points in nonlinear systems, especially suited for problems of path planning for mobile robot with nonholonomic and kinodynamic constraints. Particle Swarm Optimization is an efficient algorithm for optimization problems.

This paper presents the application of RRT combining with PSO algorithm to find a path for mobile robots in a known environment. PSO is used to search for the optimal angular and linear velocities so that the robot can move from one position to the next one on the tree. Simulation results in Matlab show that the proposed method is efficient for finding the path for mobile robot in a known environment.

**Keywords:** mobile robot; path planning; RRT algorithm; Particle Swarm Optimization

## I. GIỚI THIỆU

Robot di động ngày càng chiếm vị trí quan trọng trong các lĩnh vực nghiên cứu và ứng dụng. Một vài ứng dụng có thể kể đến như [8]: robot thăm dò sao Hoả, robot dò mìn trong quân sự, dùng trong các lĩnh vực công nghiệp và cuộc sống là robot hút bụi, robot hàn, robot kiểm tra ống ngầm.

Bài báo này giải quyết bài toán “Tìm đường

đi” cho robot di chuyển từ vị trí xuất phát đến vị trí đích một cách tối ưu. Kỹ thuật tìm đường đi theo Latombe, J.-C. [5] được hoàn thiện cho những không gian tĩnh nơi mà vị trí của tất cả các đối tượng bao hàm trong môi trường được định vị rõ ràng. Tuy nhiên kỹ thuật này có thể được áp dụng trong những không gian động như một môđun bổ sung để tránh sự va chạm với

đối tượng động.

Hai trong những phương pháp tìm đường đi phổ biến đó là phương pháp phân tích ô [7] và bản đồ đường xác suất (PRM) [5]. Phương pháp phân tích ô phụ thuộc nặng nề vào sự phân chia không gian thành các ô có kích thước. Sự phân chia đó sẽ phức tạp hơn khi robot tiếp xúc với những ràng buộc nonholonomic và kinodynamic. Trong phương pháp PRM, một đồ thị sẽ được xây dựng trong cấu hình không gian bởi việc phát thảo ngẫu nhiên những điểm và kết nối cặp những điểm cạnh nhau với một điểm của sơ đồ địa phương. Tuy nhiên, vấn đề kết nối có thể gặp khó khăn trong việc thực hiện điều khiển phi tuyến, đặc biệt là đối với ràng buộc nonholonomic và kinodynamic [3].

Phương pháp cây RRT (Rapidly-exploring Random Tree) [1], là phương pháp lập kế hoạch đường đi hiệu quả trong không gian có ràng buộc nonholonomic hơn hai phương pháp nêu trên. Nó được giới thiệu đầu tiên vào năm 1998 bởi Steve Lavalle. Cây RRT có những đặc điểm được trình bày như:

- Cây RRT là một cấu trúc dữ liệu được thiết kế cho việc tìm kiếm quỹ đạo trong không gian đa chiều
- Ngọn của cây luôn luôn được tìm thấy trong không gian có vật cản và các ràng buộc nonholonomic hoặc kinodynamic.

Vấn đề bài báo đặt ra được giải quyết như sau: thứ nhất tạo các điểm ngẫu nhiên với cây RRT nêu trên. Thứ hai dùng thuật toán PSO (tạm dịch **tối ưu bầy đàn**) [4], được Kennedy và Russell C.Eberhart giới thiệu vào năm 1995 tại một hội nghị của IEEE, để tìm vị trí và vận tốc hiện tại tốt nhất của điểm ngẫu nhiên, vị trí và vận tốc toàn cục tốt nhất.

## II. THUẬT TOÁN RRT

Robot di động có mô hình toán học như [9].

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (1)$$

Trong đó:  $(x,y)$  là vị trí của robot, góc  $\theta$  định hướng cho robot, vận tốc thẳng  $v$  và vận tốc góc  $\omega$ .

Từ mô hình toán học của robot, robot có thể di chuyển xung quanh nó một vùng trong khoảng thời gian  $\Delta t$  và nằm trong

phạm vi  $v_{max}$ . Một nguyên nhân làm robot lệch hướng là ràng buộc nonholonomic tồn tại trong môi trường chuyển động của nó.

Qua hình 1 có thể thấy robot sẽ tốn ít thời gian để di chuyển đến vùng trước và sau của nó hơn là di chuyển đến vùng bên cạnh, bởi vì phải cần một gia tốc cao hơn để lái.

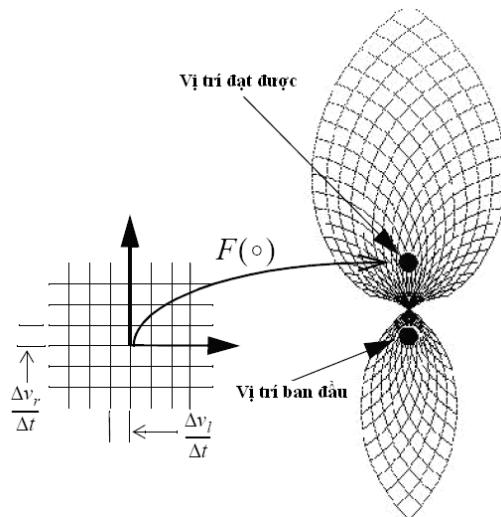
Đặt biểu thức chuyển đổi trạng thái được sử dụng với ràng buộc nonholonomic là:

$$\dot{x} = f(x, u) \quad (2)$$

Trong đó  $x$  là một cấu hình và  $\vec{u} = (v, \omega)$  được cài đặt cho ngõ vào điều khiển

Trạng thái mới:

$$x_{new} = x + f(x, u) \Delta t \quad (3)$$



Hình 1. Biểu đồ của hàm  $F(o)$ , thay đổi vận tốc bánh  $(\Delta v)/\Delta t$

để thay đổi vị trí của robot trong một đơn vị thời gian.

Đặt  $NEW\_STATE(x, u, \Delta t)$  là trạng thái mới tạo ra. Cho điểm bắt đầu ( $x_{init}$ ), cây RRT ( $\tau$ ) với  $K$  đỉnh được xây dựng như sau [2].

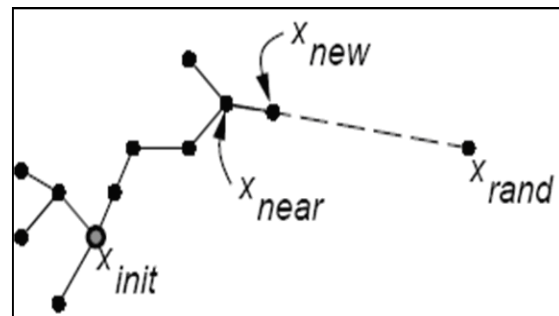
1.  $\tau \leftarrow \text{init}(x_{init});$
2. **for**  $i = 1$  **to**  $K$  **do**
3.  $x_{rand} \leftarrow \text{RAND\_STATE}();$
4.  $x_{near} \leftarrow \text{NEAREST\_NEIGHBOR}(x_{rand}, \tau);$
5.  $u \leftarrow \text{SELECT\_INPUT}(x_{rand}, x_{near});$
6.  $x_{new} \leftarrow \text{NEW\_STATE}(x_{near}, u, \Delta t);$
7.  $\tau \text{.add\_vertex}(x_{new});$
8.  $\tau \text{.add\_edge}(x_{near}, x_{new}, u);$
9. **return**  $\tau$

Trong không gian cấu hình  $C$  cho trước (bao gồm không gian tự do  $C_{free}$  không chứa vật cản và không gian vật cản  $C_{obs}$ ), cây RRT sẽ được triển khai như hình. Thứ nhất, cây RRT bắt đầu tại  $x_{init} \in C_{free}$ . Trong mỗi vòng lặp, chọn một cấu hình ngẫu nhiên  $x_{rand}$  sao cho  $x_{rand} \in C_{free}$  (bằng cách sử dụng một thuật toán tìm kiếm va chạm để loại bỏ những mẫu trong  $C_{obs}$ ). Đặt  $\rho$  là khoảng cách giữa hai cấu hình robot được xác định trên không gian cấu hình  $C$ . Bước 4, chọn cấu hình  $x_{near}$  là điểm gần tới  $x_{rand}$  nhất được tìm thấy trong một giới hạn của  $\rho$ . Bước 5 chọn một cấu hình mới  $x_{new}$ , di chuyển từ  $x_{near}$  một khoảng cách gia tăng  $\Delta t$  với sự định hướng của  $x_{rand}$ . Nếu tồn tại ràng buộc nonholonomic thì ngõ vào điều khiển  $u$  được áp dụng để kiểm soát vị trí và vận tốc tương ứng và cấu hình mới thu được theo tích phân số giữa  $(x, u)$ . Cuối cùng, một đường nối mới được thêm vào từ  $x_{near}$  đến  $x_{new}$ .

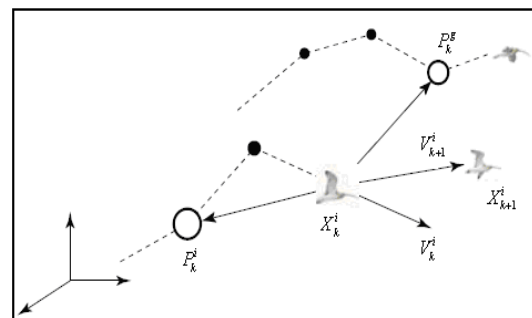
### III. THUẬT TOÁN PSO TRONG TÌM ĐƯỜNG ĐI CHO ROBOT DI ĐỘNG

PSO là một tập hợp những phần tử chuyển động với vận tốc và góc quay như hình 6 khi chúng được “ném” tự do trong không gian tìm kiếm. Mỗi phần tử có những đặc tính sau [5]:

- Vị trí hiện tại của phân tử,  $x_k^i$
- Vận tốc hiện tại của phân tử,  $v_k^i$



Hình 2. Cấu trúc cây RRT



Hình 3. Sự di chuyển của phân tử

- Vị trí tốt nhất của phân tử, liên quan tới vị trí thích hợp nhất mà phân tử đạt tới cho đến thời điểm  $k$ ,  $p_k^i$
- Vị trí tốt nhất toàn cục, liên quan tới vị trí thích hợp nhất được tìm thấy giữa các phân tử,  $p_k^g$

Trong quá trình tìm kiếm, mỗi phân tử sẽ cập nhật vận tốc và vị trí của nó để di chuyển tới vị trí hiện tại tốt nhất và vị trí toàn cục tốt nhất theo đẳng thức sau:

$$v_{k+1}^i = \omega \times v_k^i + c_1 r_1 \times (p_k^i - x_k^i) + c_2 r_2 \times (p_k^g - x_k^i) \quad (4)$$

$$x_{k+1}^i = x_k^i + v_{k+1}^i \quad (5)$$

Với  $i \in 1 \dots n$  trong  $n$  là số lượng phân tử,  $k$  số lần lặp,  $\omega$  trọng lượng quán tính,  $r_{1,2}$  số ngẫu nhiên trong khoảng  $[0,1]$  và  $c_{1,2}$  là hệ số gia tốc.

Hàm mục tiêu đánh giá tính tối ưu của phân tử là:  $f(x)$  sai số chênh lệch giữa vị trí của phân tử và vị trí hiện tại tốt nhất của nó.

Mỗi phân tử cập nhật vị trí tốt nhất của nó theo phương trình:

$$(p_k^i) \leq f(x_k^i) \begin{cases} p_k^i \\ p_{k+1}^i \end{cases} = (p_k^i) > f(x_k^i) \text{ Nếu Nếu} \quad (6)$$

Vị trí toàn cục tốt nhất của phân tử được cập nhật dựa vào phương trình:

$$p_{k+1}^g = \min(p_{k+1}^i) \quad (7)$$

Trong bài báo này, thuật toán PSO được sử dụng nhằm tìm vận tốc tối ưu để robot di chuyển giữa các điểm trên cây RRT. Vị trí và vận tốc của phân tử thứ  $i$  được mã hóa như sau:

$$p^i = (v^i, \omega^i) \quad (8)$$

$$V^i = (V_v^i, V_\omega^i) \quad (9)$$

Trong đó,  $v^i$ : vận tốc thẳng của robot,  $\omega^i$ : vận tốc góc của robot,  $V_v^i$ : tốc độ thay đổi của  $v^i$ ,  $V_\omega^i$ : tốc độ thay đổi của  $\omega^i$

#### IV. KẾT QUẢ MÔ PHỎNG

Bằng cách sử dụng phương pháp cây RRT kết hợp thuật toán PSO đã đưa ra kết quả như mong muốn là tìm được đường đi cho robot. Với các thông số cài đặt như sau:

##### Trường hợp 1:

- Thông số cài đặt cho cây RRT

Điểm bắt đầu = [25 18]; Điểm đích = [70 70 0 0 0 0 0];  $\rho=3$ ; số điểm random =  $i$ ,  $1 < i < 500$ ; cây = [25 18 0 0 0 (1-2\*rand)\*pi, 0, 0];

- Thông số cài đặt cho PSO

$w\_max=8$ ;  $v\_max=8$ ;  $\Delta T=1$ ; số lần lặp = 100; số phân tử = 20;  $c1=1.3$ ;  $c2=1.3$ .

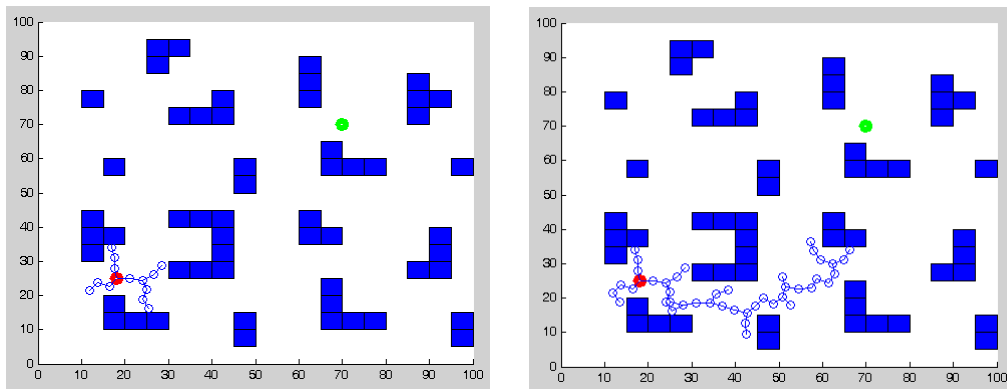
- Chú thích:

○ Điểm random

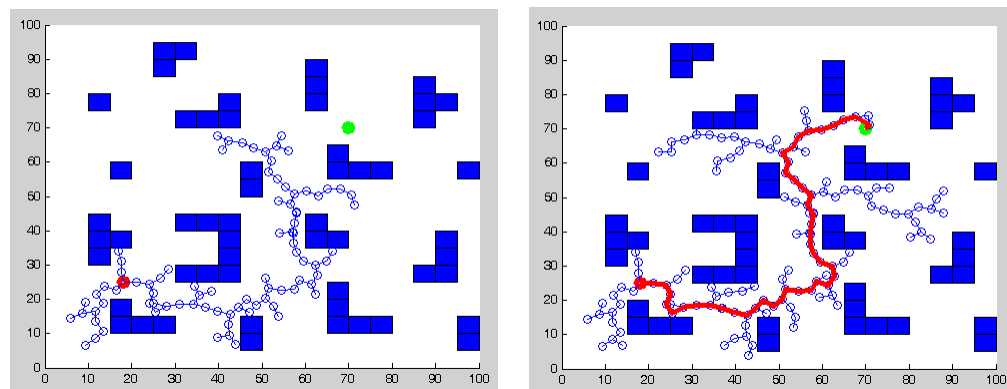
— Cây RRT

— Đường đi cho robot

Kết quả là:



Hình 4a. Giai đoạn 1 của cây RRT với  $i=14$  Hình 4b. Giai đoạn 2 của cây RRT với  $i=44$



Hình 4c. Giai đoạn 4 của cây RRT với  $i=86$  Hình 4a. Giai đoạn 4 của cây RRT với  $i=136$

## Trường hợp 2:

Không gian vật cản thay đổi, thông số cài đặt thay đổi như sau:

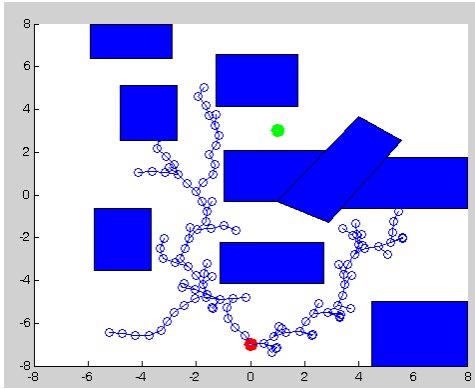
- Thông số cài đặt cho cây RRT

Điểm bắt đầu = [0 -7]; Điểm đích = [1 3 0 0 0 0 0];  $\rho=0.5$  ; số điểm random =  $i$ ,  $1 < i < 500$ ;  
 cây = [0 -7 0 0 0 (1-2\*rand)\*pi, 0, 0];

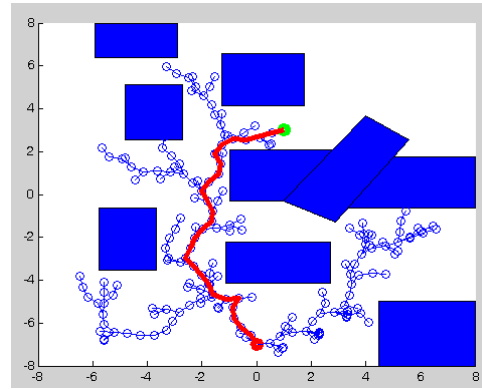
- Thông số cài đặt cho PSO

$w_{max}=1$ ;  $v_{max}=1$ ;  $\Delta T=1$  ; số lần lặp = 100; số phân tử = 20;  $c1=1.3$ ;  $c2=1.3$ .

Kết quả là:



Hình 5a. Khai triển cây RRT xét trong hệ tọa độ  $C = [-8,8] \times [-8,8]$



Hình 5b. Đường đi của robot xét trong hệ tọa độ  $C = [-8,8] \times [-8,8]$

## V. KẾT LUẬN

- Phương pháp RRT là phương pháp tìm kiếm hiệu quả trong môi trường có ràng buộc nonholonomic
- Đường đi tối ưu sẽ được tạo dựng cho robot di chuyển từ điểm bắt đầu đến điểm đích mà không va chạm vật cản.
- Khai triển các biến thể của cây RRT ứng dụng trong không gian tìm kiếm rộng lớn.

## TÀI LIỆU THAM KHẢO

Latombe, J.C, *Robot Motion Planning*. Norwood, MA, Kluwer Academic Publishers, 1991.

“Learn about Robots Includes Information on Research, Commercial and Military Robot Applications,” <http://robotics.nasa.gov/links/resources.php>

Batavia, P., Nourbakhsh, I., “Path Planning for the Cye Robot,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '00)*, Takamatsu,

Japan, November, 2000.

Dr. Jizhong Xiao, *Introduction to Robotics*, Department of Electrical Engineering City College of New York, [jxiao@ccny.cuny.edu](mailto:jxiao@ccny.cuny.edu)

<http://www.mauriceclerc.net> 29 February 2000

Lagoudakis, Michail G., *Mobile Robot Local Navigation with a Polar Neural Map*, MSc Thesis, University of Southwestern Louisiana, 1998.

Maurice Clerc ([Maurice.Clerc@WriteMe.com](mailto:Maurice.Clerc@WriteMe.com)), *Discrete Particle Swarm Optimization illustrated by the Traveling Salesman Problem*.

Roland Siegwart and Illah R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, 2004.

Russell, S., Norvig, P., *Artificial Intelligence, a Modern Approach*. Prentice Hall International, 1995.

Steven M. LaValle, *Rapidly-Exploring Random Trees: A New Tool for Path Planning*, Department of Computer Science Iowa State University Ames, IA 50011 USA, [lavalle@cs.iastate.edu](mailto:lavalle@cs.iastate.edu).