A Smooth Path Generation Approach for Sensor-based Coverage Path Planning

Alpaslan Yufka[1], Ahmet Yazıcı[2], and Osman Parlaktuna[1]

[1]Department of Electrical and Electronics Engineering, Eskisehir Osmangazi University, Eskisehir, Turkey
ayufka@gmail.com, oparlak@ogu.edu.tr
[2] Department of Computer Engineering, Eskisehir Osmangazi University, Eskisehir, Turkey
ayazici@ogu.edu.tr

Abstract

In this study, a path smoothing strategy is proposed for sensor-based coverage problems. Smooth paths are generated for the coverage problems considering mobile robot kinematics constraints. An open agent architecture-based control structure is used to implement the proposed approach on real robots. The algorithm is coded with C++ and implemented on P3-DX mobile robots in MobileSim simulation environments. It is shown that the proposed approach smoothes the curves and robot easily turns the corners. As a result of this, the completion time of the coverage decreases.

1. Introduction

Sensor-based coverage path planning problem requires every point in a given area to be covered by robot's sensors [1] as in landmine detection, foraging, and patrolling kind of applications. In sensor-based coverage problems, depending on the area, different methods may be used to model the environment. For example narrow indoor environments (where sensor range is adequate to cover every point), environment could be modeled by a Generalized Voronoi Diagram (GVD) [2]. Based on this diagram, a representative network of the environment could be constructed. Visiting all the edges of the network will be sufficient for complete coverage of this environment [3]. In other studies [4], [5], the environment is divided into robot's tool size squares to run the algorithms. Backtracking spiral algorithm (BSA) proposed in [4] is based on the execution of spiral filling paths, instead of "zig-zag" like paths used by most of the precedent algorithms. In [5], a spanning tree covering approach by subdividing the workspace into discrete cells and following a spanning tree of a graph is proposed. This algorithm uses four times of robot-tool sized squares to model the environment. In [6], the area subject to coverage is modeled with disks representing the range of sensing devices. Then the problem is defined as finding a path which runs through the center of each disk.

Most of the coverage path planning algorithms forces the robot to take sharp turns (e.g. [4], [5], [6]). These turns take significant time, since the robot must slow down, orient itself, and then accelerate. In [7] for a time-efficient coverage, number of turns was also taken into consideration to partition the coverage task among robots in a balanced manner. But the approach in [7] does not generate a smooth path. There are a few approaches for smooth path planning in coverage problems [8]. On the other hand, smooth path planning is a difficult problem since it requires trajectory generation and path planning. It is widely studied ([9], [10], [11]) for classical start-goal oriented path planning methods which differs from coverage problem.

For a differential-drive mobile robot, a smooth path may be composed of straight lines, curves, and arcs. In [12, 13], it is shown that car-like vehicles are able to reach any feasible point in the sense of the shortest path using straight and circular lines. In [14], it is shown that the Wheeled Mobile Robot (WMR) can be driven like a car-like robot if appropriate control is applied on its differential-drive wheels. Thus, by controlling translational and rotational velocities of the WMR, a straight or an arc-like desired path can be followed. In [9], [10], and [11] some methods of trajectory planning, tracking and controlling are introduced.

In this study, smooth paths are generated for the coverage problems using the circular-lines approach mentioned above. This is achieved by smoothing the initial sensor-based coverage path plans considering mobile robot kinematics. These are implemented under agent-based robot control architecture [15]. It consists of a user interface agent (UIA) which provides interaction between the architecture and the user, a planning agent (PA) which is used for generating a global path for the robot, and lastly action agent (AA) which is responsible for the robot's perception and localization. These will be explained in more details in the following sections. When the planning agent generates non-smooth coverage paths, it sends them to action agent to execute. Then, action agent firstly generates smooth paths considering robot kinematics then executes this path.

The paper is organized as follows. In the next section, some concepts on agent-based robot control architecture are introduced. The WMR's kinematics and the curvature generating are given in section 3. Section 4 is devoted to present the experimental results with a non-smooth path and a smooth-path. The paper closes with the conclusions.

2. Agent-based robot control architecture

Mobile robot applications require coordination of several modules such as perception, localization, motion planning, low-level control, human robot interface etc. Depending on the application, different robot control architectures are used to coordinate these modules. In this study, Open-Agent Architecture (OAA) [15] is used for sensor-based coverage problem. The proposed control architecture mainly consists of three agents; user interface agent, planning agent, and action agent as given in Figure 1. These agents communicate each other via a communication module (facilitator) over a network.

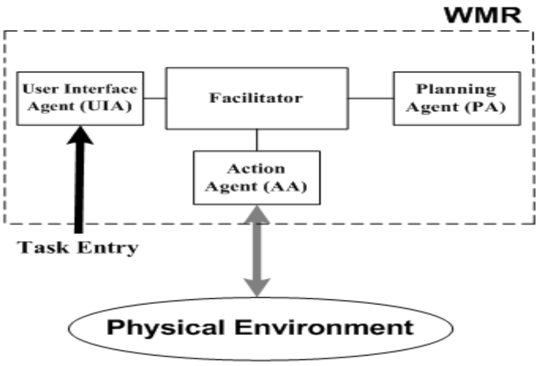


Fig. 1. Agent -based robot control architecture

The details of these agents are as follows;

User Interface Agent (UIA): This agent provides interaction between architecture and user. User initiates the starting and ending of coverage task via this agent.

Planning Agent (PA): This agent is responsible for generating sensor-based coverage path plans. The proposed planning algorithms that are given by [3, 7] are executed under this agent. These approaches are a kind of global sensor-based coverage planners, so they do not consider robot kinematics. As a result during, the execution of these paths, robots may show sharp turns. When a global plan is generated, it is sent to action agent via facilitator.

Action agent (AA): This agent is responsible for tasks related to hardware of robots (actuators and sensors). It is also responsible for robot's high-level perception and localization. The task-oriented behaviors are implemented in conjunction with the survival behaviors like obstacle avoidance by coordination of subsumption [15] approach. So, it performs behavior-based motions according to motion plans provided by PA. Differing than other studies, AA is modified to generate smooth paths using straight and circular lines.

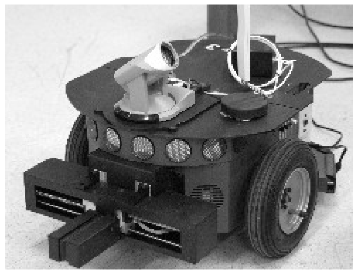
In the following subsection, the details of smoothing sensor-based coverage paths in AA are given in details.

3. The smooth path generation

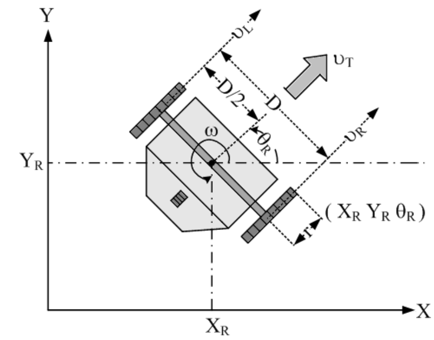
It is well known that the differential-drive mobile robot is a non-holonomic system because there are differential constraints that cannot be completely integrated [14]. To generate a smooth path for the WMR, first of all the kinematics model of the differential-drive mobile robot needs to be obtained. Using this kinematics model, a feasible curved-path is built and tracked theoretically [10].

3.1. The WMR kinematics model

Most wheeled mobile robots (WMR) have differential drives so that they are not able to move like a car-like robot. P3-DX mobile robot is an example for this kind of robot (Fig. 2 (a)) which has two main wheels attached to its motors, and a caster wheel placed in the rear. The location and orientation of the WMR body is denoted by the vector of z where the velocity vector of the WMR body is $q=[\upsilon_x\ \upsilon_y\ \omega]^t$.



a. Pioneer P3-DX mobile robot



b. Representation of velocities and robot position

Fig. 2. The WMR and its kinematics model

To construct a kinematics model of the WMR, the distance D between the left and right wheels and the wheel radius r are required. Thence, as given in equation (1) [14], υ_x and υ_y are translational velocities which are combined in υ_T as shown in Fig. 2 (b). υ_R and υ_L specify the right and left wheel velocities, and ω is the angular velocity. υ_T and υ_Ψ are interpreted as an action variable that means "translate" and "rotate" respectively if υ_T and υ_Ψ is considered as in equation (1) [14].

$$\begin{aligned} \upsilon_x &= r\frac{(\upsilon_R+\upsilon_L)}{2}\cos\theta_R \\ \upsilon_y &= r\frac{(\upsilon_R+\upsilon_L)}{2}\sin\theta_R \\ \omega &= \frac{r}{D}(\upsilon_R-\upsilon_L) \\ \upsilon_T &= \frac{(\upsilon_R+\upsilon_L)}{2} \\ \upsilon_\psi &= (\upsilon_R-\upsilon_L) \end{aligned} \tag{1}$$

Using equation (1), the kinematics model of the P3-DX can be represented as follows in equation (2).

$$\begin{bmatrix} \upsilon_x \\ \upsilon_y \\ \omega \end{bmatrix} = \begin{bmatrix} r\cos\theta_R & 0 \\ r\sin\theta_R & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \upsilon_T \\ \omega \end{bmatrix} \tag{2}$$

Based on the kinematics model, it is obvious that the location and the orientation of the WMR body is changed according to a differential driving by means of the applied voltages to the right and left wheels. If $\upsilon_R \neq \upsilon_L$, then it turns in the sense of clockwise or counter-clockwise direction respect to velocities of two wheels. In this way, this rotation generates a curvature whose center is at ICC (Instantaneous Center for Curvature) as shown in Fig. 3.

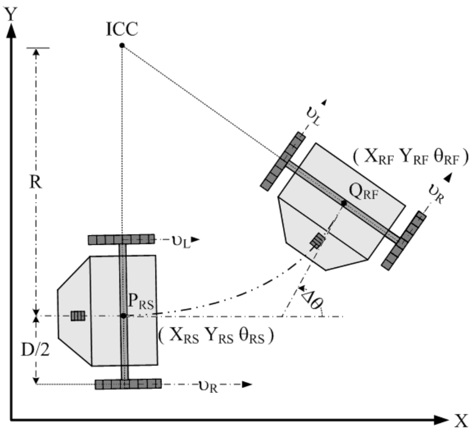


Fig. 3. A curvature trajectory

In Fig. 3, R is a radius of a curved path which is tracked by the mobile robot. The relationship between this radius of the curvature and two wheel velocities is given in equation (3).

$$R = \frac{D}{2}\left(\frac{\upsilon_R + \upsilon_L}{\upsilon_R - \upsilon_L}\right) \qquad (3)$$

If both the left and right wheel of the WMR has an equal speed ($\upsilon_R = \upsilon_L > 0$), the robot follows a straight line and goes to ∞.

3.2. Curvature and rotational trajectory

The sensor-based coverage paths generated by PA of the robot control architecture using different planning algorithms consist of straight lines as in Fig. 4. These straight lines are shaped according to the nodes. In this study, the combinations of straight lines may be one of these illustrated in Fig. 4.

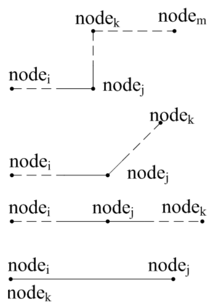


Fig. 4. Possible straight lines in the coverage problem

Then planning agent sends these possible paths to AA to follow. But, following some of these paths may cause the mobile robot to have sharp turns due to non-smooth connections at the nodes. Additionally, mobile robot must slow down, orient itself, and then accelerate. So, smoothing this path considering the robot kinematics given by equation (2) may reduce sharp turns. This results rotational trajectories which the shape depends on the location of the nodes.

Coverage plans obtained by the straight lines given in Fig. 4 may be transformed into curvature trajectories as in Fig. 5. Then, the final path is divided into three sections; before the rotation, during rotation, and after rotation. In Fig. 5, $P_S = [X_S\ Y_S\ \theta_S]^t$ is the initial point where the WMR may slow down if it is necessary and wait for the beginning of the rotation. P_{RS} is the point where the WMR starts to rotate around the point ICC, Q_{RF} is the point where the robot finishes its rotation, and $Q_F = [X_F\ Y_F\ \theta_F]^t$ is the final point where the WMR completes its tracking on the curved path. In this study, it is assumed that the length of edge that is generated by PA is equal or greater than 2L (This is satisfied in [3], [7]). Otherwise, it can be handled using other approaches [10] which is beyond the scope of this study.

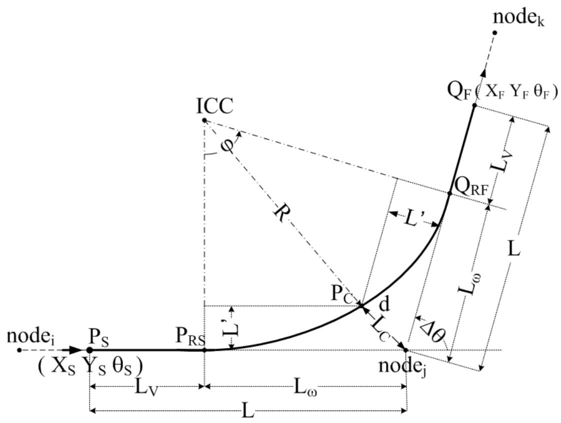


Fig. 5. The figuration of the symmetric curved-path

When the WMR is rotating from P_{RS} to Q_{RF} as illustrated in Fig. 5, the difference between the initial and the final orientation of the robot give a value of $\Delta\theta$. Since there are constraints by walls, obstacles and compelling nodes must be visited for complete coverage, the bound of path-deviation L' from the corner, through with L_C, which is related with the epsilon neighborhood of $node_j$, as defined in equation (4) [11], is considered.

$$\Delta\theta = \theta_F - \theta_S \quad or \quad \Delta\theta = \theta_{RF} - \theta_{RS}$$
$$L_c = \frac{L'}{\cos(\Delta\theta/2)} \qquad (4)$$
$$L = L_\omega + L_v$$

d is the total distance traveled by the WMR from the beginning of the rotation P_{RS} to the end of the rotation Q_{RF} with the rotation angle φ as given in equation (5) [10].

$$d = \int_{t_{P_{RS}}}^{t_{P_{RS}}+t_{Q_{RF}}} \frac{\upsilon_L + \upsilon_R}{2} dt \qquad (5)$$
$$\varphi = \frac{d}{R}$$

When the WMR reaches the point P_S, the WMR gets ready for rotating. After reaching the next point P_{RS}, the WMR rotates in a radius R centered at ICC. During the rotation, the robot updates its location and orientation frequently and controls the translational speed υ_T, rotational speed ω, the heading angle θ_R and the value of $\Delta\theta$ depends on θ_R- θ_S . If the conditions, as well as θ_R- θ_S= $\Delta\theta$, are satisfied and the robot reaches the point Q_{RF}, then the WMR terminates its rotation and goes to the desired and final point Q_F as shown in Fig. 5. The robot reduces its translational velocity depending on the radius R of the curvature.

4. Implementation for the coverage

The curvature tracking for the WMR is coded in C++ and tested using a P3-DX robot in the MobileSim's simulation environment. The P3-DX robot shown in Fig. 2 (a) has an onboard P3-800 computer with Linux OS. The sensors on the robot are: SICK LMS Laser range finder, Sonar sensors, Camera, Compass. A wireless network is set to communicate with other robots and computers. Mainly SICK LMS Laser range sensor is used for the coverage task. The sensor has normally a range of 50 meters, but for experimental purposes the range is restricted to 3 meters with software.

Fig. 6 shows a topological map of the test platform and GVD-based network considering 3 meters range in ESOGU artificial intelligence and robotic laboratory. Covering all the area is achieved by following all the edges in this figure.

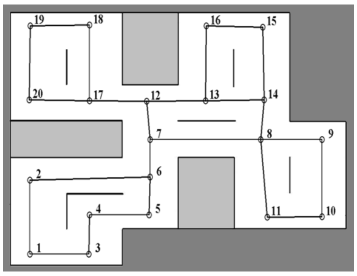


Fig. 6. GVD-based network of the test environment and map

For this purpose, PA generates the motion plan using the approach in [3] and sent to AA. The WMR's tours are as follows; **(1-2), (2-6), (6-7) , (7-8) , (8-9) , (9-10) , (10-11) , (11-8) , (8-14) , (14-15) , (15-16) , (16-13) , (13-14)** , *(14-13)* , **(13-12) , (12-17) , (17-18) , (18-19) , (19-20) , (20-17)** , *(17-12)* , **(12-7)** , *(7-6)* , **(6-5) , (5-4) , (4-3) , (3-1)**. The mobile robot performs coverage task while passing through bold-written edges and use the italic-written edges for passing. A straight line combination of this plan is given in Fig.7.

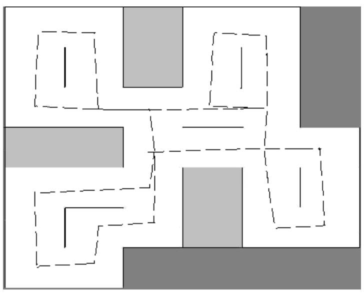


Fig. 7. The combinations of straight line

Applications are executed in two ways. In the first experiment, AA follows the path plan coming from PA without non-smoothing. In the second experiment, AA smoothes the path and then follows. The AA normally follows this path as in the next section.

4.1. The application of non-smooth path

During the experiment, the AA sets the WMR'S translational speed 200 mm/s and the WMR decelerates, stops and accelerates at each node. The P3-DX robot covers the given area as illustrated in Fig. 8 in MobileSim simulation environment. Note that, there is no curvature tracking at predefined nodes which must be visited by the WMR for complete coverage. In addition, in this experiment, nodes force the WMR to visit each of them in the epsilon-neighborhood which is a predefined constant determined by the user so that the robot has a non-smooth path when it reaches the node and sets its head angle to the next node, because it has no ability to track a smooth-path. For this experiment, the total path traveled by the WMR is 41713.574899 mm (**41.71 meters**) and the task completion time **297 seconds** (4.57 minutes).

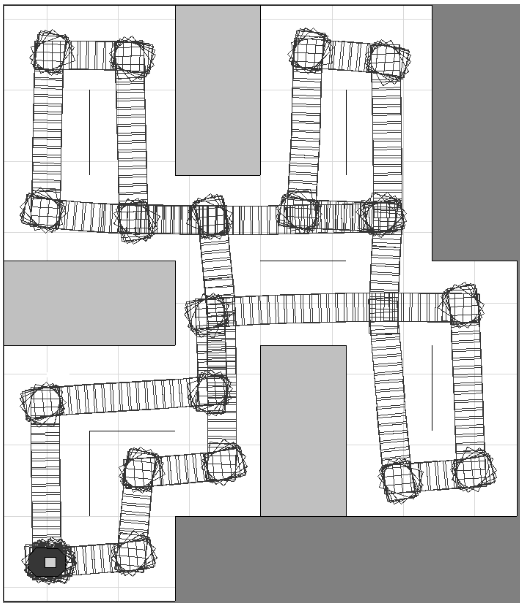


Fig. 8. Non-smooth path tracked by the WMR

4.2. The application of smooth path

In the second application, the P3-DX mobile robot starts initially from node 1 and it follows the same tour. In this experiment, the WMR's translational speed is 200 mm/s on a straight line and 150 mm/s on a curved line. But, the only difference between the second and first experiment is that the second experiment includes a curvature tracking when the WMR approaches to the nodes to be visited. The mobile robot performs coverage task with a smooth path as shown in Fig. 9. In addition, the total path travelled by the WMR is 40238.863750 mm (**40.24 meters**) and the task completion time **233.5 seconds** (3.54 minutes).

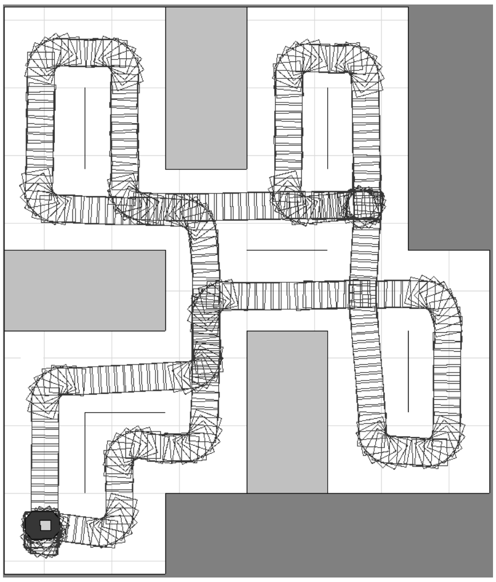


Fig. 9. Smooth path tracked by the WMR

5. Conclusion and Proposals

A path smoothing strategy is proposed for sensor-based coverage problems. This is achieved by considering mobile robot kinematics. An open agent architecture-based control structure is used to implement the proposed approach on real robots. The proposed approach have decreased the coverage time and smoothed sharp turns during the experiments. Thus, a smooth path generating eliminates frequent stops at nodes which cause unnecessary acceleration and deceleration. The improvement on the total completion time compared the previous study are pointed out.

6. Acknowledgement

This work is supported by the Scientific and Technical Research Council of Turkey (TUBITAK), Contract No 107E064, project title: "Mobile Robot Route Planning for Complete Coverage of Dynamic Indoor Environments".

7. References

[1] E.U. Acar, H. Choset, and J.Y. Lee, "Sensor-based coverage with extended range detectors", *IEEE Transactions on Robotics*, v.22 (1), 2006, pp.189-198.

[2] E. Cinar, O. Parlaktuna, and A. Yazici, "Küçük Ortamların Tam Kapsanması Amacıyla Genelleştirilmiş Voronoi Diyagramı Modellemesi", *Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu*, 2008, p.115-118.

[3] A. Sipahioğlu, A. Yazıcı, G. Kirlik, and O. Parlaktuna, "A sensor-based coverage approach in Dynamic Environments", *6th International Symposium on Intelligent and Manufacturing Systems*, October 14-16, 2008, pp. 502-509.

[4] E. Gonzalez, O. Alvarez, Y. Diaz, C. Parra, and C. Bustacara, "BSA: A Complete Coverage Algorithm", *IEEE International Conference on Robotics and Automation*, 2005, pp. 2040 – 2044.

[5] Y. Gabriely and E. Rimon, "Spanning-tree based coverage of continuous areas by a mobile robot", *Annals of Mathematics and Artificial Intelligence*, 2001, pp.77–98.

[6] M. Kapanoglu, M. Ozkan, A. Yazıcı, and O. Parlaktuna, "Pattern-Based Genetic Algorithm Approach to Coverage Path Planning for Mobile Robots", G. Allen et al. (Eds.): ICCS 2009, Part I, LNCS 5544, pp. 33–42, 2009.

[7] M. Ozkan, A. Yazici, M. Kapanoglu, and O. Parlaktuna, "A genetic algorithm for task completion time minimization for multi-robot sensor-based coverage", *the 24th IEEE International Symposium on Intelligent Control (ISIC)*, SaintPetersburg, Russia, 2009, to be presented.

[8] S. Garrido, L. Moreno, D. Blanco, and F. Martin "Smooth path planning for non-holonomic robots using fast marching*", IEEE International Conference on mechatronics*, 2009, pp. 1 – 6.

[9] F. Lamiraux and J. P. Laumond, "Smooth motion planning for car-like vehicles", *IEEE Transactions on Robotics and Automation*, 2001, 17 (4): pp. 498-502.

[10] S. Hana, B. Choia, and J. Lee, "A precise curved motion planning for a differential driving mobile robot", *the International Federation of Automatic Control (IFAC)*, Volume 18, Issue 9, 2008, Pages 486-494.

[11] C. H. Kim & B. K. Kim, "Minimum-Energy Translational Trajectory Generation for Differential-Driven Wheeled Mobile Robots", *Journal of Intelligent and Robotic Systems*, Volume 49, Number 4, pp. 367-383, August, 2007.

[12] L. E. Dubins, "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents," *Amer. J. Math.*, vol. 79, pp. 497–516, 1957.

[13] J. A. Reeds and R. A. Shepp, "Optimal paths for a car that goes both forward and backward," *Pacific J. Math.*, vol. 145, no. 2, pp. 367–393, 1990.

[14] S. M. LaValle, "Planning Algorithms", Cambridge University Press, 2006.

[15] M. Özkan, A. Yazıcı, and O. Parlaktuna, "Erkin Tabanlı Gezgin Robot Kontrol Mimarisinin Ortam Kapsanması Probleminde Kullanımı", *Otomatik Kontrol Ulusal Toplantısı*, Bildiriler Kitabi İstanbul, 2008, pp. 909-914.

[16] R. R. Murphy, "Introduction to AI Robotics", MIT Press, 2000.