PATH PLANNING FOR INTELLIGENT WHEELCHAIR BASED ON MODIFIED TENTACLES METHOD

A Thesis Presented
By
Jingwei Liu
to
The Department of Electrical & Computer Engineering

in partial fulfillment of the requirements
for the degree of

Master of Science
in the field of

Electrical & Computer Engineering

Northeastern University
Boston, Massachusetts

December 2017
ABSTRACT

In this research, a path planning method for differential drive intelligent wheelchair is described. This method is derived from a path planning algorithm called driving with tentacles. The approach of this research is applying wheelchair’s state, generating a modified tentacle trajectory that decrease the driving deviation between the selected path and the wheelchair’s true path. The selection of the tentacle (a drivable path) relies on a heuristic cost function for each tentacle, and other variables such as tentacle’s length, distance to the destination, wheelchair’s speed and collision condition with obstacles. MATLAB simulation and practical implementation on Permobil C300 wheelchair with ROS (Robot Operating System) are also provided in this paper.
ACKNOWLEDGMENTS

Studying at Northeastern University for the past three years has been an unforgettable memory for me. In here, I improved my skill of study and increased a great deal of knowledge.

First, I would like to express my gratitude to my advisor, Professor Taskin Padir, whose expertise, patience and support throughout my studies added considerably to my graduate experience. I really appreciate his vast knowledge and his assistance in my writing report.

Furthermore, I must also acknowledge my committee members, Professor Deniz Erdogmus and Professor Dagmar Sternad. I am grateful for their taking time out from their busy schedules and their suggestions in my defense.

I would also thank my family for the support they provided in my entire life and in particular, I must acknowledge my mother, without whose love and encouragement, I would not have finished this thesis.

At last, I would thank all of my friends who helped me in past three year. Thank you.
TABLE OF CONTENTS

LIST OF TABLES.................................................................................................................................vi
LIST OF FIGURES...............................................................................................................................vii
LIST OF SYMBOLS...............................................................................................................................ix
1. INTRODUCTION.....................................................................................................................................1
2. PROBLEM STATEMENT..........................................................................................................................4
  2.1 Introduce of “Driving with Tentacles” .............................................................................................4
  2.2 Modified Driving with Tentacles ....................................................................................................5
  2.3 The Experiment Platform: Powered wheelchair ..............................................................................6
3. WHEELCHAIR MODEL.........................................................................................................................8
  3.1 Kinematic Model ...........................................................................................................................8
  3.2 Position and Steering Angle .........................................................................................................9
  3.3 Speed .............................................................................................................................................10
  3.4 Distance from Destination and Path Following Time .......................................................................11
4. GENERATION OF MODIFIED TENTACLES...................................................................................13
  4.1 Path Generation of Steady State ..................................................................................................13
  4.2 Path Generation of Transient State ..............................................................................................14
  4.3 Zoom-in Rate of Steady Path Trajectory ......................................................................................15
  4.4 Comparison of Modified Tentacles Method and Driving with tentacles ........................................16
5. TENTACLE TRAJECTORY SELECTION..........................................................................................21
  5.1 Prerequisite Selection ..................................................................................................................21
    5.1.1 Prerequisite Selection in Boundary Map ..............................................................................21
    5.1.2 Prerequisite Selection in Obstacle Map ..................................................................................23
LIST OF TABLES

2.1 Technical specifications of Norma wheelchair.................................................................7
3.1 Table of speed set number and value.............................................................................11
7.1 Table of biggest deviation in each test...........................................................................39
LIST OF FIGURES

2.1 Driving with tentacles simulation and tentacle indices in speed set 0 ........................................ 5
2.2 Two sequential tentacles and wheelchair driving deviation ....................................................... 6
2.3 Technical specifications of Norma wheelchair ............................................................................ 7
3.1 Differential drive robot kinematics .............................................................................................. 9
4.1 Wheelchair yaw rate for the jth number of tentacle in speed set k ............................................. 15
4.2 Tentacle pattern shape of different Zoom-in Rates .................................................................... 15
4.3 Parameters setting of the experiment wheelchair ....................................................................... 16
4.4 Driving with tentacles pattern and modified tentacle pattern in speed set 0 and
    their comparison. Because the speed is low, the difference between the patterns
    of two methods is small ............................................................................................................... 18
4.5 Driving with tentacles pattern and modified tentacle pattern in speed set 4 and
    their comparison. The longer the transient state lasts, the bigger the difference
    between the patterns of two methods is .................................................................................. 19
4.6 Driving with tentacles pattern and modified tentacle pattern in speed set 8 and
    their comparison. The longer the transient state lasts, the bigger the difference
    between the patterns of two methods is .................................................................................. 20
5.1 Simulation of prerequisite selection in road map condition ......................................................... 22
5.2 Simulation prerequisite selection in sidewalk map condition ..................................................... 23
5.3 One simulation of the prerequisite selection in obstacle map condition ................................... 24
5.4 Another simulation of the prerequisite selection in obstacle map condition ............................ 25
5.5 Simulation of the final path selection .......................................................................................... 26
5.6 Complete path planning working flow chart

6.1 A sketch map of the simulation result

6.2 Result of modified tentacle path planning in speed set 0

6.3 Result of driving with tentacles path planning in speed set 0

6.4 Result of modified tentacle path planning in speed set 4

6.5 Result of driving with tentacles path planning in speed set 4

6.6 Result of modified tentacle path planning in speed set 8

6.7 Result of driving with tentacles path planning in speed set 8

7.1 Corridor map created by LIDAR

7.2 Visualization in Rviz

7.3 X-axis and Y-axis positions comparison between selected path and true path using Modified tentacle method

7.4 X-axis and Y-axis positions comparison between selected path and true path using Driving with tentacles method
# LIST OF SYMBOLS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>j</td>
<td>The tentacle number</td>
</tr>
<tr>
<td>k</td>
<td>The tentacle set number</td>
</tr>
<tr>
<td>ICC</td>
<td>Instantaneous Center of Curvature of wheelchair</td>
</tr>
<tr>
<td>ω</td>
<td>Angular velocity of the wheelchair</td>
</tr>
<tr>
<td>V_&lt;sub&gt;l&lt;/sub&gt;</td>
<td>Left wheel velocity of the wheelchair</td>
</tr>
<tr>
<td>V_&lt;sub&gt;r&lt;/sub&gt;</td>
<td>Right wheel velocity of the wheelchair</td>
</tr>
<tr>
<td>x</td>
<td>X coordinate of wheelchair’s center of two wheels</td>
</tr>
<tr>
<td>y</td>
<td>Y coordinate of wheelchair’s center of two wheels</td>
</tr>
<tr>
<td>Θ</td>
<td>Wheelchair’s steering angle</td>
</tr>
<tr>
<td>V&lt;sub&gt;k&lt;/sub&gt;</td>
<td>Wheelchair linear velocity in set k</td>
</tr>
<tr>
<td>V&lt;sub&gt;s&lt;/sub&gt;</td>
<td>Wheelchair minimum speed in simulation</td>
</tr>
<tr>
<td>V&lt;sub&gt;e&lt;/sub&gt;</td>
<td>Wheelchair maximum speed in simulation</td>
</tr>
<tr>
<td>(te&lt;sub&gt;x&lt;/sub&gt;, te&lt;sub&gt;y&lt;/sub&gt;)</td>
<td>coordinates of tentacle path end point</td>
</tr>
<tr>
<td>(dest&lt;sub&gt;x&lt;/sub&gt;, dest&lt;sub&gt;y&lt;/sub&gt;)</td>
<td>coordinates of wheelchair’s destination</td>
</tr>
<tr>
<td>d&lt;sub&gt;g&lt;/sub&gt;</td>
<td>Euclidean distance between tentacle end point and destination</td>
</tr>
<tr>
<td>l&lt;sub&gt;selected&lt;/sub&gt;</td>
<td>Length of selected each tentacle path</td>
</tr>
<tr>
<td>t&lt;sub&gt;path&lt;/sub&gt;</td>
<td>Time used by following selected path</td>
</tr>
<tr>
<td>l&lt;sub&gt;j&lt;/sub&gt;</td>
<td>The length of the jth tentacle</td>
</tr>
<tr>
<td>r&lt;sub&gt;j&lt;/sub&gt;</td>
<td>The radius of the jth tentacle</td>
</tr>
<tr>
<td>̇ω(t)</td>
<td>Rotation acceleration of the wheelchair</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Zoom-in rate</td>
</tr>
<tr>
<td>PR</td>
<td>The plotting resolution of the jth tentacle in set k</td>
</tr>
</tbody>
</table>
Chapter 1

INTRODUCTION

Robotics has grown very fast in the past two decades with the advance in research and enabling technologies. Especially ground vehicles, has been the focus of extensive research effort. For unmanned ground vehicles, path planning is an important field that lets vehicles find the shortest or otherwise optimal path between two points. The optimal paths could be paths that minimize the time consumption, the amount of turning or whatever a specific application requires. Therefore, extending research focus on this field and aim to advance the unmanned vehicle systems.

Heuristic-based path planning methods like A* can achieve shortest or minimum time consumption path between two points [1, 2]. Dynamic A* (D*) is applied in partially known environment and can re-plan a new path based on the previous path and the information of environment [3, 4]. Because A* and D* become computational expensive when the search space is large due to high dimension of the map, a more recent development known as Rapidly-Exploring Random Trees (RRTs) addresses the path planning problem by using a randomized approach that aims at quickly exploring a large area of the search space [5]. Also, a Bi-RRTs method, an improved version of RRTs connects two trees which take start point and goal point as their roots respectively to achieve a faster coverage in the searching space [6, 7]. There are other path planning methods which can satisfy both path following and obstacle avoidance. Driving with tentacles is first introduced for four-wheel vehicles which computes the drivable path as a combination of pre-calculated circular arcs [8]. Because of the fast-to-implement property of the driving with tentacles method, many research aim to extend on it.

For driving with tentacles method based research, one of the focuses lies on improving the tentacle selection process to reduce its computational complexity. One improvement in this area
is implementing fuzzy-logic constructs which simplifies the selection process. This method uses three factors such as speed, distance to the desired position and closeness to the obstacle to create the fuzzy member functions to help select the final path [9]. Another contribution provides a ripple tentacles motion planning method. In their research, it creates many concentric circular ripples that spreading around in all direction to check each tentacle’s drivable length. By choosing the tentacle which has the longest drivable length, the method can avoid frequent adjustment in weighing factors in different environment [10]. There are also some research focus on improving the tentacle trajectory generation to get additional obstacle avoidance ability. In [11, 12], they utilize the global route and local environment data to achieve obstacle avoidance by generating a finite number of tentacles to the left or the right side with different offset shifts, while remaining substantially parallel to the base path. Also, some research use multiple shape of tentacles to form the drivable path that is more realistic in real road structure. In [13, 14] the shape of the tentacles is clothoid and is determined by speed and steering angle. Moreover, some research combine different implementations to improve the tentacle method. One of those methods provides a Markov Decision Process-based implementation to fast perform path planning [15].

In assistive technology, lots of effort has been put on developing robotic technologies for intelligent wheelchairs. Early efforts in the development of smart wheelchairs tackle the issue by providing the user with an external switch or button to trigger a change in the mode of operation [17, 18]. A semi-autonomous wheelchairs can not only augment the user inputs by providing additional obstacle avoidance or wall following functionalities, but also can achieve a semi-autonomous navigation which combines an adaptive motion control and a SLAM(Simultaneous localization and mapping) algorithm [19].

Inspired by these methods, our research introduce a modified tentacles path planning algorithm implemented on differential drive autonomous wheelchair. For differential drive vehicles, their angular velocity is related to the wheels’ speed and the length of the center of two wheels. However, for steer-wheel drive vehicles, their angular velocity is related with speed, steering wheel direction, vehicle’s speed direction and its previous angular velocity. Compared to
steer-wheel drive vehicles, differential drive vehicle’s dynamic has less complexity. Hence, in our approach, we will take the wheelchair’s dynamic state into consideration to compute each modified tentacle. Also, we will form a heuristic cost function to help choose the final path in tentacle selection process. In contrast with driving with tentacles method, the proposed modified tentacles path planning method decreases the deviation between the selected path and the wheelchair’s true path when wheelchair following a selected path.

The paper is organized as follows: In Chapter 2, we will describe the problem in detail. Chapter 3 will present the model of the wheelchair. Chapter 4 demonstrates how to apply wheelchair dynamic model to compute the paths in our modified tentacles method. In Chapter 5, we will introduce a path selection algorithm using heuristic function. And Chapter 6 presents the simulation results and comparison with driving with tentacles algorithm. Chapter 7 provides an implementation on Permobil C300 wheelchair. Finally, Chapter 8 is the conclusion.
Chapter 2

PROBLEM STATEMENT

Our research is concerned with path planning for autonomous wheelchairs. To get a more precise path, we will use mathematical method to calculate the path. The computer will hold-up the calculation speed so the running wheelchair will receive data in time. Therefore, a completed and succinct mathematical method is needed if we want to achieve a good and complete path planning system.

2.1 Introduction of “Driving with Tentacles”

In the Paper ‘Driving with tentacles: Integral Structures for Sensing and Motion’ [8], the paths (tentacles) are split into 16 sets of 81 circular arcs because the circular arcs can reasonably well approximate all possible clothoids the vehicle can drive. Each set depends on vehicle’s linear speed. Each arc represents a possible vehicle trajectory and the length of each arc varies depending on the tentacle number $j$ and the tentacle set number $k$. In Figure 2.1, a) shows the simulation of tentacle set 0 and b) shows its tentacle number index in [8]. The simulation of tentacle Set 0 is the sketch of the set number 0 of paths of the wheelchair, (0, 0) is the start position of the wheelchair. For other tentacle Sets, the tentacle indices obey the same forming rules like Set 0.
2.2 Modified Driving with Tentacles

In section 2.1, we introduce the Driving with Tentacle method. However, in Figure 2.2 a), when the wheelchair follow two sequential tentacles, the wheelchair will eventually deviate from the considered destination because its yaw rate cannot change suddenly. In Figure 2.2 b), it shows the wheelchair switches from tentacle number 0 to number 68 in Set 0. The wheelchair’s final trajectory is the red path.

In our research, in order to make the wheelchair’s final path closer to the selected path, we utilize wheelchair’s state (linear velocity and angular velocity) to modify the generation of tentacles. By checking wheelchair’s state such as current speed and yaw rate, we can use mathematical representation of the wheelchair’s position to form a trajectory that represent the transient state. Then, when the wheelchair’s yaw rate becomes steady, the driving with tentacles method can be applied.
2.3 The Experiment Platform: Powered Wheelchair

In order to test the effectiveness of the proposed method, the method has been implemented on real wheelchair platform (named Norma: Northeastern Robotic Mobility Assistant). Figure 2.3 shows the experiment wheelchair used as a test bed for this research with the LIDAR implemented, which will detect the surrounding obstacles.

Norma is a differential drive wheelchair that only the two front wheels provide driving force. Robot Operating system (ROS) is installed in Norma and the packages we created in experiment are in standard ROS form. Our test implementations also have the portability of implementing the algorithm to other differential drive robots which have feedback of linear velocity and angular velocity of the robots. Table 2.1 shows the technical specifications of our Norma wheelchair.
Table 2.1 Technical specifications of Norma wheelchair

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>L x W x H</td>
<td>915 x 635 x 1500</td>
<td>mm</td>
</tr>
<tr>
<td>Ground clearance</td>
<td>89</td>
<td>mm</td>
</tr>
<tr>
<td>Maximum user weight</td>
<td>120</td>
<td>kg</td>
</tr>
<tr>
<td>Max speed</td>
<td>5</td>
<td>m/s</td>
</tr>
<tr>
<td>Range</td>
<td>30-35</td>
<td>km</td>
</tr>
<tr>
<td>Kinematic</td>
<td>Differential Drive</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Chapter 3

THE MODEL OF THE WHEELCHAIR

In this chapter, we mainly discussed about the robot model and its mathematical representations. The wheelchair model represents the motion of the wheelchair as the following parameters: wheelchair’s position, velocity, steering angle, distance between wheelchair and destination and the time used by following a tentacle trajectory. Also, the model has the following properties:

I. The model is two-dimensional (2-D).

II. The forces that effect the motion (such as friction, air resistance) are not taken into consideration in the model.

III. The model deals with the geometry model of the map.

3.1 The Kinematic Model of Wheelchair

Because the rear wheels of the wheelchair do not provide driving force, we simply consider the wheelchair as a common differential-drive robot. That is the wheelchair consists of 2 drive wheels mounted on a common axis, and each wheel can independently being driven either forward or backward.

Figure 3.1 shows the differential drive robot kinematic from [16]. In the figure, ICC is known as Instantaneous Center of Curvature, the point the robot rotate about. $V_l$ and $V_r$ are the left and right wheel linear velocity along the ground. $l$ is the distance between the centers of the two wheels and $\omega$ is the angular velocity and $R$ is the signed distance from the ICC to the midpoint between the wheels. $\theta$ is the angle between the robot moving direction and the X-coordinate.
Because the rate of rotation $\omega$ about the ICC must be the same for both wheels, we can write the following equations:

\[
\begin{align*}
\omega(R - l/2) &= V_l \\
\omega(R + l/2) &= V_r \\
\omega &= \frac{(V_r - V_l)}{l}
\end{align*}
\]  

Therefore, according to equation 3.1 to equation 3.3, at any instance, by knowing $l$, $V_l$, and $V_r$. We can calculate $\omega$ and $R$ at any time. After we have this differential drive robot kinematic, we can form the parameters in path planning next.

### 3.2 Position and Steering Angle of Wheelchair

In general, we can describe the position of a robot capable of moving in a particular direction $\theta(t)$ at a given velocity $V(t)$ as:
\[
x(t) = \frac{1}{2} \int_0^t V(t) \cos[\theta(t)] \, dt \quad (3.4)
\]
\[
y(t) = \frac{1}{2} \int_0^t V(t) \sin[\theta(t)] \, dt \quad (3.5)
\]
\[
\theta(t) = \int_0^t \omega(t) \, dt \quad (3.6)
\]

In our case, when applying equation 3.3, the position and steering angle can be calculated as:

\[
x(t) = \frac{1}{2} \int_0^t [V_r(t) + V_l(t)] \cos[\theta(t)] \, dt \quad (3.7)
\]
\[
y(t) = \frac{1}{2} \int_0^t [V_r(t) + V_l(t)] \sin[\theta(t)] \, dt \quad (3.8)
\]
\[
\theta(t) = \int_0^t \omega(t) \, dt \quad (3.9)
\]

where \( \omega(t) \), \( V_l(t) \) and \( V_r(t) \) are the time-varying form of parameters introduced in section 3.1. The position and steering angle will be used to create the wheelchair’s pose information in simulation and experiment.

### 3.3 The Speed of Wheelchair

From reference \([8]\), the formula to calculate linear velocity is shown in equation 3.10. In Chapter 4, we will use the wheelchair’s speed along with other parameters to help us generate the tentacle path.

\[
V_k = V_s + q^{12} (V_e - V_s) \quad (3.10)
\]

Where \( V_s \) is the lowest speed \( V_s = 0.25m/s \) and \( V_e \) is the highest speed \( V_e = 10m/s \), with \( q = \frac{k}{15} \). The motivation for the exponential factor \( q^{12} \) is to sample low speeds more frequently. Hence, based on equation 3.10, we can calculate wheelchair’s velocity for all speed
sets. The speed set number and speed value is shown in Table 3.1.

Table 3.1: Table of speed set number and value

<table>
<thead>
<tr>
<th>Speed Set Number</th>
<th>Speed value (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-3</td>
<td>0.250 0.628 1.119 1.663</td>
</tr>
<tr>
<td>4-7</td>
<td>2.246 2.859 3.467 4.157</td>
</tr>
<tr>
<td>8-11</td>
<td>4.836 5.532 6.244 6.670</td>
</tr>
<tr>
<td>12-15</td>
<td>7.710 8.462 9.225 10</td>
</tr>
</tbody>
</table>

We need to note that in our wheelchair platform, the linear velocity varies from 0 m/s to 5 m/s. But for the concern of this method’s portability, we still consider the condition that linear velocity varies from 0 m/s to 10 m/s.

3.4 The Distance from Destination and Path Following Time

In this section, we will use two variables \(dg\) and \(t_{path}\) to represent the distance between wheelchair’s each endpoint and its destination and time used by following the path. We use Euclidean distance to calculate \(dg\). \(t_{path}\) will be calculated by summing the time used by following each selected tentacle trajectory. The equations is show below:

\[
dg = \sqrt{(te_x - dest_x)^2 + (te_y - dest_y)^2)} \quad (3.11)
\]

\[
t_{path} = \Sigma \frac{l_{selected}}{v_k} \quad (3.12)
\]

where \(te_x\) and \(te_y\) represent the x and y coordinate of each tentacle end point. \(dest_x\) and \(dest_y\) represent the x and y coordinate of destination. \(l_{selected}\) represents the length of each selected tentacle path. One thing to note here is the variables \(dg\) and \(t_{path}\) will be used in Chapter 4 and Chapter 5.
In this chapter, we illustrate the details of the model of the wheelchair and some valuable parameters which are essential in forming the modified tentacles generation process and in modified tentacles selection process. In next chapter, we will first present how to generate the modified tentacles.
Chapter 4

THE GENERATION OF MODIFIED TENTACLE

In chapter 2, we noted that because of the yaw rate cannot change suddenly, there is a transient state before the wheelchair can get its steady yaw rate when following two sequential tentacle trajectories that have different index numbers. Therefore, a trajectory to show the wheelchair’s true path should be the path of transient state combines the path of steady state.

In this chapter, we will describe the method used to calculate the paths of transient state and steady state. Also, we will show the comparison of this method when compared with the driving with tentacles method.

4.1 The Path Generation of Steady State

The driving with tentacles method details the geometry of the tentacles which are circular arcs emanating from the robot’s center of turning used as we mentioned in the Chapter 2, including the tentacle set number \( k \) from 0 to 15, the tentacle’s number \( j \) from 0 to 80, the radius of each tentacle \( r_j \), the length of each tentacle \( l_j \). Then unit of \( r_j \) and \( l_j \) is meter.

In this section, equation 4.1 and 4.2 calculate \( r_j \) and \( l_j \) of the \( j \)th tentacle in set \( k \) respectively.

\[
\begin{align*}
    r_j &= \begin{cases} 
    (1.15)^j \times \left( \frac{(8+33.5 \times \left( \frac{k}{15} \right)^{1.2}}{1.2 \times \left( \frac{8}{2} \right)^{1.2} \times (1-\left( \frac{k}{15} \right)^{0.9})} \right) & | j = 0, \ldots, 39 \\
    (1.15)^{8-j} \times \left( \frac{(8+33.5 \times \left( \frac{k}{15} \right)^{1.2}}{1.2 \times \left( \frac{8}{2} \right)^{1.2} \times (1-\left( \frac{k}{15} \right)^{0.9})} \right) & | j = 40, \ldots, 80 \\
    \infty & | j = 40 
    \end{cases} \\
    l_j &= \begin{cases} 
    8 + 33.5 \times \left( \frac{k}{15} \right)^{1.2} + 20 \times \frac{j}{40} & | j = 0, \ldots, 40 \\
    8 + 33.5 \times \left( \frac{k}{15} \right)^{1.2} + 20 \times \frac{8+j}{40} & | j = 41, \ldots, 80
    \end{cases}
\end{align*}
\]
In equation 4.1, the reason for the exponential form is to have more tentacles with small curvatures and a coarser distribution at larger curvatures. However, there is no physical reason for this choice and a uniform distribution of the radius would probably work, too. In equation 4.2, the constants 8 and 33.5 indicate that for different speed set $k$, the tentacles should have a minimum length that provide a sufficient look-ahead distance. Therefore, when the wheelchair’s state becomes steady, we can always use equation 4.1 and 4.2 to calculate the path that the wheelchair will follow.

4.2 The Path Generation of Transient State

In this section, we will illustrate the process of generating the transient path. To generate the transient path, first we need to know the wheelchair’s steady yaw rate for each tentacle index number in each speed set. Because in each speed set, the speed will remain constant, so, by using equation 4.1 along with equation 3.10, we can calculate the steady yaw rate as:

$$\omega = \frac{V_k}{r_j}$$ (4.3)

Figure 4.1 shows the steady yaw rate and we should note that counter-clockwise direction has the positive value. In speed set 5, at tentacle index number 0 and 80, the wheelchair will get its maximum yaw rate around 0.2 rad/s.

Thus, the steering angle in equation 3.9 can be changed to equation 4.4, where $\dot{\omega}(t)$ is the rotation acceleration of the wheelchair.

$$\theta(t) = \int_0^t \left[ \frac{V_k}{r_j} - \dot{\omega}(t) \right] dt$$ (4.4)

To simplify the calculation, we assume that the rotation acceleration $\dot{\omega}(t)$ is a constant value. Hence, when the two sequential tentacle trajectories have different index number, we can easily use equation 4.4 along with equation 3.7 and equation 3.8 to calculate the position and steering angle of the wheelchair during its transient period.
4.3 Zoom-in Rate of Steady Path Trajectory

In this section, we introduce a parameter zoom-in rate $\beta$ to change the length of the path trajectory in steady state. The effect of this parameter is when the wheelchair approaches to its destination, reducing the size of the overall tentacle shape to let the wheelchair find a closer end point to the destination. Equation 4.5 shows the calculation of $\beta$, where $\sigma$ is a positive scalar less than one and $dg$ is introduced in section 3.4. $dg_{max}$ is the distance of start position and destination.

$$\beta = \sigma + (1 - \sigma) \cdot \frac{dg}{dg_{max}}$$

(4.5)

Figure 4.2 shows from No.40 in speed set 0, when $\beta = 1$ and $\beta = 0.5$, the shape of the tentacle pattern. We can find the bigger $\beta$ is, the larger size of the pattern is.

Figure 4.2 Tentacle pattern shape of different Zoom-in Rates
4.4 Comparison of Modified Tentacles Method and Driving with Tentacles

In this section, we will use MATLAB to show the differences between the modified tentacle method and the driving with tentacles method. Because we need to plot in MATLAB, we first need to set a plotting resolution $PR$ (pixel per meter). This resolution is defined by us and equation 4.5 shows the calculation of $PR$. And in the simulation part and experiment part, we will still use this plotting resolution to generate the path.

$$PR = \left\lfloor \frac{1000}{V_k} \right\rfloor$$ (4.6)

We will use speed set 0, 4 and 8 as examples. In these examples, the rotation acceleration value is 0.2 rad/s$^2$ and the zoom-in rate is 1. The reason to choose such a value is because in our pre-test, by varying the parameters shown in the red box in Figure 4.3, we recorded the time used to make the wheelchair get its steady speed. In the pre-test, we found a higher acceleration would cause people who sit on the wheelchair feel uncomfortable. In the meanwhile, too small the acceleration would make the wheelchair use too much time to arrive the steady state. Therefore, we found the acceleration value between 0.1 $\sim$ 0.3 rad/s$^2$ is reasonable, which means in the parameters shown in Figure 4.3, the parameters value is between 15 $\sim$ 45. So, in our examples, we choose 0.2 rad/s$^2$ as the rotation acceleration.

Figure 4.3 Parameters setting of the experiment wheelchair
In Figure 4.4, a) shows the driving with tentacles pattern in speed set 0. We can find the tentacle pattern is symmetric and whatever the previous tentacle index number is, the driving with tentacles will always use one tentacle pattern based on the speed. However, b) and c) shows different tentacle pattern in modified tentacle method. In b) the previous selected tentacle is No.0. Thus, this pattern is no longer symmetric and all tentacles except No.0 have slight changes when compared with the driving with tentacles pattern in d). In c) the previous selected tentacle is No.40, so, this pattern is still symmetric but is also different when compared with driving with tentacles pattern in e).

In speed set 0, despite differences between driving with tentacles pattern and modified tentacle pattern, the differences are very small because the speed and yaw rate of the wheelchair is both very small and when the wheelchair comes to steady state, the wheelchair doesn’t move far away from its initial position. However, in Figure 4.5 and 4.6, the patterns in speed set 4 and 8 changes dramatically because when the wheelchair comes to its steady state, it already drives a long distance from its initial position.

Figure 4.5 shows the patterns and comparison in speed set 4, we can find in c) that those tentacles which have opposite direction with the previous selected tentacle have more deviation. Even in d), higher speed makes the modified tentacle pattern change a lot when compared with the driving with tentacles pattern.

Figure 4.6 shows the patterns and comparison in speed set 8, this figure shares similar characteristics with Figure 4.4.

We should note if the two sequential tentacles have the same index number, there is no changes between using driving with tentacle method or modified tentacle method. We can find this property through each d) in Figure 4.4, 4.5 and 4.6. When two sequential tentacle index number are 0, the tentacle path of index number 0 match perfectly.
Figure 4.4 Driving with tentacles pattern and modified tentacles pattern in speed set 0 and their comparison. Because the speed is low, the difference between the patterns of two methods is small.
Figure 4.5 Driving with tentacles pattern and modified tentacles pattern in speed set 4 and their comparison. The longer the transient state lasts, the bigger the difference between the patterns of two methods is.
Figure 4.6 Driving with tentacles pattern and modified tentacles pattern in speed set 8 and their comparison. The longer the transient state lasts, the bigger the difference between the patterns of two methods is.
Chapter 5

TENTACLE TRAJECTORY SELECTION

In real path planning systems, after generating the tentacle path candidates, the path planning algorithm also needs to provide a feasible path selection method. In this chapter, the tentacle path selection process is separated into two parts. One is prerequisite selection which will examine whether the tentacle path candidates meet some prerequisite requirement such like road only, sidewalk only or examine whether the path is blocked by some obstacles. Another part is final path selection that chooses the best path among those which are already selected by the first part. Also, at the end of this chapter, we will provide the overall structure of the path planning algorithm.

5.1 Prerequisite Selection

The reason to have a prerequisite selection is because this part can reduce the computation in tentacle selection process. Among those different circumstances in the map, we category them into two classes: boundary map and obstacle map. In boundary map, wheelchair needs to follow the path have one or two boundaries and avoid obstacles without touching the edge of the prerequisite setting boundaries. In obstacle map, wheelchair needs to find a path from start position to its destination without colliding with obstacles. The key point of this part of selection is eliminating all tentacle path that counter with obstacles or boundaries.

MATLAB simulation will be used to help illustrate the selection process in this chapter.

5.1.1 Prerequisite Selection in Boundary Map

In this section, we will show two types of selection details in boundary map situation. In Figure 5.1, we show the road map condition. The blue star is the start position of the wheelchair and the two black bars are the boundaries of road. The red tentacle arcs are eliminated for the
reason that they are all out of edge of the two boundaries. The green tentacle arcs are feasible path trajectories that can be selected in the final path selection process. Figure 5.2 shows the sidewalk condition which have only one boundary. And the upper area of the black bar is the drivable area.

Figure 5.1. Simulation of prerequisite selection in road map condition
5.1.2 Prerequisite Selection in Obstacle Map

In Figure 5.3 and 5.4 we show the prerequisite selection in obstacle map condition. In the figures, there are ten obstacles which are circles or rectangles randomly located in different areas in the map. The radius of the circles vary from 5 to 10 meters. The length and width of the rectangles vary from 5 to 10 meters. The blue start is the start point of the wheelchair. The red tentacle arcs are eliminated because they collide with obstacles and the green arcs are feasible.
path trajectories that can be selected in the final path selection process.

Figure 5.3 One simulation of the prerequisite selection in obstacle map condition
5.2 Final path selection

The final path selection executes sequentially after the prerequisite selection. Because there are still multiple feasible paths after prerequisite selection shows in Figure 5.1 to 5.4, we need to choose the optimal one. In this section, we will form a heuristic function to help us to create a standard to choose the optimal path. Each of the tentacle path candidate in this selection process will get a value based on the heuristic function. In our case, we will choose the tentacle path which
has the smallest value as the optimal path.

The heuristic function has two factors: $dg$ and $t_{path}$ which we have introduced in section 3.4. Equation 5.1 shows the calculation of the heuristic function where $\alpha_1$ and $\alpha_2$ are parameters that can be used to change the two factors at a gross level.

$$h = \alpha_1 \cdot pt + \alpha_2 \cdot t_{path}$$ (5.1)

In figure 5.5, we show a simulation of the Final path selection. In this simulation, we choose speed set 0 with parameters $\alpha_1 = 4$ and $\alpha_2 = 1$. The obstacle number is seven. The blue star is the start position of the wheelchair and the red star is its destination. After the final path selection, among those green arcs, the blue arc has the minimal heuristic function value.
5.3 Overall Structure of the Path Planning Algorithm

After we introduce the modified tentacle generation and the tentacle selection method, we can combine them into a complete path planning algorithm. Figure 5.6 shows the working flow chart of the complete path planning algorithm. At the beginning of this path planning, we need to provide input data: chosen speed set, start position, destination and map which hold the information of map condition and obstacles. Then we will check whether the destination is valid. If yes, the algorithm will get into loop. First, getting wheelchair’s state to help calculate the modified tentacle candidates. After this, using prerequisite selection to eliminate those tentacles that collide with boundaries and obstacles. Then, calculate the left tentacle candidates’ heuristic function value and choose the path which has the smallest heuristic function value as the optimal path. The following is plotting the path on map and checking whether the end point of the path reaches the destination. If yes, loop ends and exit, if not, compute a zoom-in rate and loop returns to the process of getting wheelchair’s state and repeats. We should also note that if all possible paths are eliminated by the prerequisite selection, a warning will be shown and the algorithm ends.
In this chapter, we mainly discuss the tentacle selection method and the complete path planning algorithm working flow. In next chapter, we will use MATLAB to create a simulation to test our path planning algorithm.
Chapter 6

SIMULATION AND PERFORMANCE COMPARISON

In this chapter, we will show simulation results using our modified tentacle method. In the simulation, we examine the performance that evaluating the deviation when the wheelchair follow the path generated by modified tentacle method in random obstacles map without any re-planning. Also, we show a performance comparison between modified tentacle method and driving with tentacles method.

6.1 Simulation Test bed and Parameters

Our simulation is based on MATLAB R2016b. In simulation, map is created by function BinaryOccupancyGrid in Robotic Toolbox. We also use interface ExampleHelperRobotSimulator to create a differential drive robot.

Some main parameter setting in our simulation is shown below:

1. Rotation Acceleration Rate $\dot{\omega}(t) = 0.2 \text{ rad/s}^2$
2. Zoom-in Rate Coefficient $\sigma = 0.4$
3. Arrival Check tolerance = 5m
4. Map Resolution = 50 pixels/meter
5. Robot Control Rate = 30Hz

6.2 Performance and Comparison

In this section, we show simulation in speed set 0, 4 and 8. For all simulated speed sets, we only show successful simulations. Those simulation with destination invalid we will not discuss in our paper.
Figure 6.1 shows a sketch map of the simulation results. The arrow in red circle represents the simulated wheelchair, the black arc represents the selected final path and the blue arc represents the wheelchair’s true path when it following the black arc.

6.2.1 Result and Comparison of Speed Set 0

Figure 6.2 shows one successful path planning from the start position to destination in speed set 0 using modified tentacle method. In a), blue star (20, 25) represents the start position, black start represents the final path end position and red star (65, 55) represents destination. Red arcs are blocked arcs and the blue arc is the final selected path. In b), we can find that wheelchair true path almost coincide with the selected final path. In c) and d), we show the X-axis and Y-axis
position of selected path (Blue) and simulated path (Red). Also, by computing the deviation between selected path and simulated path according to c) and d), the biggest deviation is nearly 0.1 meter.

Figure 6.2 Result of modified tentacle path planning in speed set 0.

Figure 6.3 shows the path planning simulation in same settings using driving with tentacles method in speed set 0. In a), like Figure 6.2, each drawing has the same definition. In b) we can find at the beginning, the wheelchair can follow the path accurately. However, in the final of the path following, deviation becomes obvious. This result fit our finding in section 4.3 that there are small differences in tentacle patterns whether concern the wheelchair’s state or not. But as the
slight differences accumulate, the influence of differences will grow bigger. In c) and d), we show the X-axis and Y-axis position of selected path (Blue) and simulated path (Red). In addition, the biggest deviation is nearly 0.9 meter.

![Figure 6.3](image)

**Figure 6.3** Result of driving with tentacles path planning in speed set 0.

### 6.2.2 Result and Comparison of Speed Set 4

Figure 6.4 shows one successful path planning from the start position to destination in speed set 4 using modified tentacle method. In a), like Figure 6.2, each drawing has the same definition. In b), we can find that the wheelchair still follow the path accurately. In c) and d), we show the X-
axis and Y-axis position of selected path (Blue) and simulated path (Red). And the biggest deviation is around 0.4 meter.

Figure 6.4 Result of modified tentacle path planning in speed set 4.

Figure 6.5 shows the path planning simulation in same settings using driving with tentacles method in speed set 4. In a), like Figure 6.2, each drawing has the same definition. In b), we can find at the beginning, the deviation grows because the wheelchair start to turn right. However, because of the second turn is opposite to the first one, the deviation reduces in the middle part of the true path. In c) and d), we show the X-axis and Y-axis position of selected path (Blue) and simulated path (Red). We can also find c) and d) fit what we illustrated in b). Last, the biggest
deviation is around 0.9 meter.

![Figure 6.5 Result of driving with tentacles path planning in speed set 4.](image)

**6.2.3 Result and Comparison of Speed Set 8**

Figure 6.6 shows one successful path planning from the start position to destination in speed set 8 using modified tentacle method. In a), like Figure 6.2, each drawing has the same definition. In b), we can find that the wheelchair follow the path accurately. In c) and d), we show the X-axis and Y-axis position of selected path (Blue) and simulated path (Red). The biggest deviation is 0.3 meter.
Figure 6.6 Result of modified tentacle path planning in speed set 8.

Figure 6.7 shows the path planning simulation in same settings using driving with tentacles method in speed set 8. In a), like Figure 6.2, each drawing has the same definition. In b), we can find the deviation first grows then decreases. The reason is only at the beginning a transient state occurs, then the following trajectory is all in steady state. In c) and d), we show the X-axis and Y-axis position of selected path (Blue) and simulated path (Red). The biggest deviation is 4 meter in the middle part of the true path.
Figure 6.7 Result of driving with tentacles path planning in speed set 8.
Chapter 7

EXPERIMENT RESULTS

After the MATLAB simulation, we implement both driving with tentacles method and our modified tentacles method on Norma wheelchair which embedded with Robot Operating System (ROS). Using C++ and Python, we create a global planner which generate the tentacle way points and a local planner that provide an obstacle avoidance function. Figure 7.1 shows the corridor map which is created by our mounted LIDAR and we will test our algorithm in this corridor. In our experiment, we set the wheelchair’s speed to 0.25m/s. For both driving with tentacles method and our modified tentacles method, we test their performance between the same start position and destination for 4 times.

Figure 7.2 shows the visualization when wheelchair following the path in Rviz. The green square is the footprint of the wheelchair. The purple arc is the selected path. Figure 7.3 shows the X-axis position and Y-axis position of true path using modified tentacle method and the positions of selected path in the second test. The left one shows X-axis position and the right one shows the Y-axis position. Figure 7.4 shows the X-axis position and Y-axis position of true path using driving with tentacles method and the positions of selected path in the second test. The left one shows X-axis position and the right one shows the Y-axis position. For Figure 7.3 and Figure 7.4, the blue arc represents the selected path and the red arc represents the wheelchair’s final true path.

We can find the biggest deviation in Figure 7.3 is 0.18 meter and the biggest deviation in Figure 7.4 is 0.40 meter. Table 7.1 shows the biggest deviation in all 4 test. We can find that the biggest deviation of using our modified tentacles method is almost half of the biggest deviation of using driving with tentacles method. Therefore, without re-planning, using modified tentacle method can reduce the driving deviation.
Figure 7.1 Corridor map created by LIDAR

Figure 7.2 Visualization in Rviz
Figure 7.3 X-axis and Y-axis positions comparison between selected path and true path using modified tentacle method

Figure 7.4 X-axis and Y-axis positions comparison between selected path and true path using driving with tentacles method

Table 7.1: Table of biggest deviation in each test

<table>
<thead>
<tr>
<th>Biggest deviation [m]</th>
<th>Test 1</th>
<th>Test 2</th>
<th>Test 3</th>
<th>Test 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driving with tentacles method</td>
<td>0.35</td>
<td>0.4</td>
<td>0.37</td>
<td>0.42</td>
</tr>
<tr>
<td>Modified tentacles method</td>
<td>0.2</td>
<td>0.18</td>
<td>0.22</td>
<td>0.24</td>
</tr>
</tbody>
</table>
Chapter 8

CONCLUSIONS

In this paper, we proposed a path planning algorithm for ground wheelchair based on a modified tentacle method. The approach of this method is when generating the tentacle trajectories, we take the wheelchair’s dynamic into consideration. With this improvement, we can reduce the re-planning computation that caused by driving deviation as providing the wheelchair a more precise path to follow. The algorithm was successfully implemented both in MATLAB simulation and a real wheelchair platform, and in both instances the wheelchair achieved a less driving deviation compared with implementing a driving with tentacles method. In simulation, we successfully illustrate the speed sets, tentacle generation and tentacle selection process. Also, we provided three different speed sets to show performance in modified tentacle method and driving with tentacles method. In real experiment, we implemented the modified tentacle method into our Norma wheelchair. The experimental results shows by using the modified tentacle method, the driving deviation reduced. In the meanwhile, there are still several aspects can be improved in our algorithm. One of those improvements is considering speed acceleration or deceleration when wheelchair follows the path. In this situation, when computing the modified tentacle trajectories, we need to add the speed change factor in our model. Another interesting idea is all possible tentacles are blocked by obstacles. In this situation, it is easy to let the wheelchair achieve a pivot turning and re-calculate the possible tentacles. Finally, our algorithm is in 2D space. But it is possible to extend to 3D by taking the tentacle generation, tentacle selection and the slope of surface into consideration.
REFERENCES


