Design and implementation of a new speed planner for semiautonomous systems

Mustafa DEMİR1,*, Volkan SEZER2
1Department of Mechatronics Engineering, Graduate School of Science Engineering and Technology, Istanbul Technical University, Istanbul, Turkey
2Department of Control and Automation Engineering, Faculty of Electrical and Electronics Engineering, Istanbul Technical University, Istanbul, Turkey

Received: 11.08.2017 • Accepted/Published Online: 14.12.2017 • Final Version: 30.03.2018

Abstract: In this paper, a new speed planning method is developed for semiautonomous systems by improving our previous fuzzy logic-based approach. In this proposed method, an extended risk factor is calculated on top of the classical risk factor, using environmental factors and the user’s speed reference. We obtain safer speed values in critical scenarios with this new extended risk factor definition. Another improvement comes from the design of the semiautonomous architecture. Instead of the fully autonomous solution of the previous work, we calculate the final speed reference by combining the risk factor-based speed value and user’s speed reference, which provides a semiautonomous solution. The developed fuzzy logic-based semiautonomous speed planner was tested in simulations and real environments on a differential drive wheelchair platform controlled via head movements. The results of the tests show that the proposed fuzzy-based semiautonomous speed planner using an extended risk factor provides safer transportation than its previous variant.

Key words: Speed planner, speed control, semiautonomous systems, fuzzy logic, wheelchair

1. Introduction

The speed planning of autonomous and semiautonomous vehicles is as important as obstacle avoidance in terms of safety. Obstacle avoidance methods enable the vehicle to reach the target without collision in unknown environments. When examined from the view of speed planning, obstacle avoidance methods can be divided into 2 groups: with and without a speed planner. The dynamic window approach (DWA) [1] is one of the obstacle avoidance methods that includes a speed planner. In this method, collisionless angular and linear speed pairs are calculated, and the pair that maximizes the objective function is selected for motion. However, there are cases where the DWA fails to provide vehicle orientation. Studies aimed at developing this method are given in [2,3].

In the artificial potential field (APF) [4], the target position is defined as an artificial attractive potential field, while the obstacles are defined as artificial repulsive potential fields. These fields generate forces that push or pull the robot. The robot slows down due to the repulsive force of the obstacle as the vehicle moves towards the obstacle. Therefore, in the APF, the speed of the vehicle varies as the vehicle moves towards the target point. However, the APF has a very important problem, the local minimum, and there are studies that have sought to solve this [5,6]. In addition to these methods, the curvature velocity method [7] and velocity obstacles [8] are speed planner-including obstacle avoidance methods.

*Correspondence: demirmust@itu.edu.tr
Obstacle avoidance methods that do not include a speed planner, such as bug algorithms [9], vector field histograms [10], the follow-the-gap method (FGM) [11], and the obstacle restriction method [12], determine only the vehicle’s orientation and need an external speed planner. For this reason, additional speed planners should be used with these obstacle avoidance methods. In most cases, speed planners determine the speed reference according to the curvature of the road and the vehicle dynamics [13,14]. On the other hand, the conditions of the environmental factors, such as obstacle density, minimum distance values to obstacles, etc., are very important for autonomous and semiautonomous vehicles’ speed reference determination.

Several intelligent wheelchair studies that are based on shared autonomy can be found in the literature. In [15], the user provides only direction commands, and the speed planning is performed by a neural network-based obstacle avoidance algorithm. Hence, the user does not have any control over the speed of the wheelchair. In another recent study [16], the user gives both the direction and velocity commands to the wheelchair. However, a potential field method that does not need an external speed planner is used as the obstacle avoidance algorithm in [16]. Another shared-control wheelchair platform is presented in [17], which is controlled by brain signals. In this study, the user provides high-level commands that are converted to direction and speed references by path-planning and path-tracking algorithms. The user does not have direct control of speed as in [15]. Another shared-autonomy strategy is shown in [18] for semiautonomous robots, and it calculates the speed using DWA without external effort.

In [19], fuzzy-based desired velocity is planned using the minimum distances to the obstacles and the goal angle. In [20], the criterion for vehicle speed determination minimizes lateral acceleration, which is mostly related to vehicle dynamics. In [21], speed is planned using both vehicle dynamics and path properties.

In this paper, a new fuzzy logic-based semiautonomous speed planning method is developed, which is the extended version of a previously designed planner for autonomous vehicles [22,23]. In the previous approach, only distances to the obstacles and the angle of the obstacles were considered for risk factor calculation. In this study, we extend these parameters by adding the user’s speed reference for risk factor calculation. By doing so, we make the wheelchair’s response more restrained for the user’s high speed references. We show that this redefinition of the risk factor (namely, extended risk factor) provides safer solutions compared to the previous approach. The real tests show that the developed approach works reliably in real time. In addition, the user’s speed reference and the calculated speed reference are fused by the weighting strategy based on the extended risk factor. Semiautonomous adaptation of previous fuzzy logic-based speed planning, which was designed for autonomous robots, is another contribution of this study.

The outline of the paper is as follows: Section 2 illustrates the technical approach for fuzzy speed planning, the developed fuzzy speed planning, and semiautonomous adaptation subsections. Section 3 gives information about the simulation environment, experimental platform, experimental tests, and comparisons. Hardware implementation and real experimental tests are illustrated in Section 4. Finally, conclusions are given in Section 5.

2. Technical approach

2.1. Fuzzy speed planning for autonomous systems

In the fuzzy-based speed planning method, which was developed for autonomous systems [22,23], the speed of the system is adjusted by calculating the risk coming from the obstacles in the environment. The risk factor is calculated for each obstacle with the help of fuzzy logic, by using obstacle positions within the vehicle’s field of view. Among the calculated risk factors, the maximum risk factor specifies the final speed of the vehicle.
If an obstacle is very near the vehicle and stands in the same direction as the vehicle, the risk factor becomes very large and the speed of the vehicle must be reduced. If the obstacle is far away or in another direction, the risk factor is not that high, which results in relatively higher speed value decisions. The fuzzy rules are defined from this perspective.

The fuzzy logic-based decision maker’s inputs are the distance and angle values of obstacles, while the output is the risk factor. These inputs are fed to a Mamdani-type fuzzy inference system in which triangular fuzzy membership functions are used. These membership functions are composed of 5 fuzzy sets. The fuzzy sets of the distances to the obstacles ($d_{min}$) are defined as very near (VN), near (N), medium (M), far (F), and very far (VF). The fuzzy sets for the angle of the obstacles ($\Phi_{obs}$) are defined as far left (FL), left (L), on front (OF), right (R), and far right (FR). The fuzzy sets of the risk factor ($rf$), which is the output of the system, are defined as very small (VS), small (S), medium (M), large (L), and very large (VL). The speed planner structure for autonomous systems is shown in Figure 1.

The distances of the obstacles are normalized to be in the range of [0,1], while the angle values of the obstacles are normalized to be in the range of [-1,1]. The output risk factor takes continuous values in the interval of [0,1]. The rules for the fuzzy block are illustrated in Table 1, and the fuzzy surface obtained based on the generated rules is illustrated in Figure 2.

Using Eq. (1), a linear speed reference is obtained for the autonomous systems due to the risk factor, which is calculated using fuzzy logic:
2.2. Fuzzy speed planning for semiautonomous systems

The speed planning method, which determines the linear speed of a vehicle by using fuzzy logic according to the risk factor of the environment, was designed for autonomous vehicles [22,23]. Speed planning for semiautonomous systems must combine both the speed reference obtained by the risk factor and the speed reference that is given by the user. For this reason, we have developed a semiautonomous speed planning strategy in this paper. Through these two speed values, Eq. (2) calculates the final speed reference:

\[ V_{out} = K_u V_{user} + K_f V_{fuzzy}. \] (2)

\[ V_{out} \]: Linear speed of the vehicle.
\[ V_{user} \]: Linear speed reference given by user.
\[ V_{fuzzy} \]: Linear speed reference determined using fuzzy.
\[ K_u \]: Weight coefficient of \( V_{user} \).
\[ K_f \]: Weight coefficient of \( V_{fuzzy} \).

\( K_u \) and \( K_f \) are weight coefficients used for combining the risk factor-based and user-defined speed reference values. In this study, we calculate these weight factors directly from the risk factor itself. The \( K_u \) and \( K_f \) coefficients are calculated as given in Eq. (3) and Eq. (4):

\[ K_f = rf, \] (3)

\[ K_u = 1 - K_f. \] (4)

If the risk factor calculated by the fuzzy logic is high, the speed reference calculated by the fuzzy \( V_{fuzzy} \) is low; for safety, the weight coefficient of this speed reference \( K_f \) is automatically high. Thanks to the risk factor, safer movement is obtained by assigning more weight to the low speed value calculated by fuzzy logic. When the risk factor of the environment is low, the weight coefficient user’s speed reference \( K_u \) is high. In this situation, since the risk factor is low, the effect of the user on final speed automatically increases. Additionally, when the user’s speed reference is smaller than the risk factor-based value, the system takes only the user’s decision regardless of the risk factor. The semiautonomous speed planner structure is shown in Figure 3.

2.3. Extended fuzzy speed planning for semiautonomous systems

The use of distances to the obstacles and obstacle angle values for speed planning is insufficient in some situations. If the speed reference of the user is too high, it poses a collision risk to the system, especially in obstacle-dense areas, even if it is combined with the classical risk factor definition. Therefore, the previously calculated risk factor should be reinterpreted by considering the user’s speed reference. In this study, in addition to the previously demonstrated fuzzy-based semiautonomous speed planner, the user’s speed reference is also considered in the calculation of the new risk factor. We call this the extended risk factor (erf), which is calculated by two cascaded Mamdani-type fuzzy inference systems as illustrated in Figure 4.

The user speed reference is normalized to be in the range of \([0,1]\). The fuzzy sets of the user speed reference (\( V_{user} \)) are defined as very slow (VS), slow (S), medium (M), fast (F), and very fast (VF).
First, the risk factor is calculated in the Fuzzy-1 block. As is understood from its inputs and output, the Fuzzy-1 block is the same as the previously defined Fuzzy block in Figure 1 and Figure 3. The risk factor, which is the output of the Fuzzy-1 block, is the input of the Fuzzy-2 block together with the user’s speed reference. The output of the Fuzzy-2 block is the extended risk factor. The fuzzy sets of the extended risk factor are defined as very low (VL), low (L), medium low (ML), little low (LL), very little low (VLL), medium (M), very little high (VLH), little high (LH), medium high (MH), high (H), and very high (VH). The rules for the Fuzzy-2 block are shown in Table 2, and the fuzzy surface obtained based on the generated rules is illustrated in Figure 5.

<table>
<thead>
<tr>
<th>$rf$</th>
<th>$V_{user}$</th>
<th>VS</th>
<th>S</th>
<th>M</th>
<th>F</th>
<th>VF</th>
</tr>
</thead>
<tbody>
<tr>
<td>FS</td>
<td>VL</td>
<td></td>
<td>L</td>
<td>ML</td>
<td>LL</td>
<td>VLL</td>
</tr>
<tr>
<td>S</td>
<td>L</td>
<td>ML</td>
<td>LL</td>
<td>VLL</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>LL</td>
<td>VLL</td>
<td>M</td>
<td>VLH</td>
<td>LH</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>MH</td>
<td>MH</td>
<td>H</td>
<td>H</td>
<td>VH</td>
<td></td>
</tr>
<tr>
<td>VL</td>
<td>H</td>
<td>H</td>
<td>VH</td>
<td>VH</td>
<td>VH</td>
<td></td>
</tr>
</tbody>
</table>

The weights of the speed references vary according to the extended risk factor. The $K_u$ and $K_f$ coefficients are calculated as given in Eq. (5) and Eq. (6):

$$K_f = erf,$$  \hspace{1cm} (5)

$$K_u = 1 - K_f.$$.  \hspace{1cm} (6)

3. Simulations and comparison

3.1. Simulations

The simulation platform is a differential-drive wheelchair. A computer-aided design (CAD) model of the wheelchair was prepared by using SOLIDWORKS as shown in Figure 6. In order to use the wheelchair in a realistic Gazebo environment [24], the CAD model was exported to Universal Robot Description File (URDF) format. The speed planner algorithms were simulated in the Gazebo environment using a robot operating system (ROS) [25]. The Gazebo simulation environment is shown in Figure 7, with the wheelchair and the
obstacles around it. In order to achieve the desired angular and linear velocities, Eq. (7) and Eq. (8)) [26] were utilized.

\[ v = \frac{v_R + v_L}{2} \text{ (m/s)} \]  

\[ w = \frac{v_R - v_L}{L} \text{ (rad/s)} \]  

The wheelchair’s rear wheel radius \( r \) is 0.2 m and the distance between the 2 rear wheels \( L \) is 0.595 m. While the linear velocity of the system is determined by the speed planning method, the angular velocity is calculated by our recently developed obstacle avoidance approach, the improved follow-the-gap method (FGM-I) [27]. The overall diagram of the system is illustrated in Figure 8.

In FGM-I, gaps between the obstacles in the robot’s field of view are determined first. The gap that maximizes the utility function given in Eq. (9) is then selected from these gaps, and the final heading angle is calculated using Eq. (10). Using FGM-I, the system is able to reach the target by following both safe and short trajectories. More information about FGM-I can be found in [27].

\[ U_n = k_1d_{\text{gap,n}} + k_2(\pi - \Phi_{\text{gap,n-to-goal}}) \]  

\( U_n \): Utility function value of nth gap.
\( d_{\text{gap,n}} \): Size of the nth gap.
\( k_1 \): Weight coefficient for the gap size.
\( \Phi_{\text{gap,n-to-goal}} \): Angle between nth gap center and goal point.
\(k_2\) : Weight coefficient for angle.

\[
\Phi_{final} = \frac{\alpha \Phi_{goal} + \Phi_{gap,c}}{d_{min} + 1}
\]  

(10)

\(\Phi_{final}\) : Final heading angle.
\(\Phi_{gap,c}\) : Gap center angle.
\(\Phi_{goal}\) : Goal point angle.
\(d_{min}\) : Minimum distance to obstacles.
\(\alpha\) : Safety coefficient.

According to the diagram given in Figure 8, a risk factor is calculated using the distance and angle values of the obstacles, which are detected by a LIDAR sensor. The extended risk factor is calculated using the risk factor, which is derived from the environment and the user's instantaneous speed reference. The final heading angle is calculated by the FGM-I, which combines the user-defined goal angle, the closest distance to obstacles, and the midpoint of the selected gap, as mentioned previously.

A simulation environment is prepared to test and show the achievements of the new speed planner. The designed Gazebo environment includes the wheelchair model, wall-shaped obstacles, and cylinders of various sizes, as illustrated in Figure 7.

A virtual camera is placed on the system, which provides a more realistic field of view from the robot’s perspective. The images provided from the camera are visualized with the Rviz visualization tool, and the users control the system through this visualization. A camera image that was obtained during the test using the Rviz visualization tool is shown in Figure 9.

In addition, a virtual force field-based speed planner (VFFSP) [28] is simulated in order to demonstrate the gains of the newly proposed method. The VFFSP aims to slow down the vehicle when it moves towards an obstacle. In the VFFSP, the x-component of the repulsive force, \(x_F\), has a reducing effect on velocity. The final speed reference is calculated using Eq. (11):

\[
V_{out} = V_{user} \times \exp \left\{ - \sum x_F \right\}
\]  

(11)

\(V_{out}\) : Linear speed of the vehicle.
$V_{user}$: Linear speed reference given by user.

$x_F$: x-component of the repulsive force.

As shown in Eq. (11), when the x-component of the repulsive force is zero, there is no change to the user speed reference. Conversely, when the x-component of the repulsive force tends to infinity, the final speed reference approaches zero. More information about this method can be found in [28].

3.2. Comparison

In the prepared simulation environment, fuzzy speed planner (FSP), extended fuzzy speed planner (EFSP), virtual force field based speed planner (VFFSP), and without-speed planner (WSP) scenarios were tested. Tests were conducted with 20 users and users tested each method twice in the simulation environment. In total, 40 simulations were obtained for each method. Figure 10 shows one of the tests.

During the tests, the starting point was determined to be [0-0], while the corridor exit was defined as the goal region. The trajectories of the wheelchair are shown for each method separately. The simulation results of WSP, VFFSP, FSP, and EFSP are illustrated in Figures 11, 12, 13, and 14, respectively. During all of the tests, FGM-I was used as the obstacle avoidance method. In the WSP tests (Figure 11), it was observed that some users could not adjust the linear speed of the wheelchair and thus hit the obstacles. Among the 40 tests, 3 collisions were detected in this mode. Collision points are shown as red circles.

In using the VFFSP, the planning was not sufficient to adjust the linear speed of the wheelchair. Therefore, the trajectories passed very close to obstacles. One collision was observed, as shown in Figure 12.

On the other hand, the wheelchair arrived at the goal region without collision in FSP support, as shown in Figure 13. Although there were no collisions during the 40 tests, the trajectories sometimes passed very close to obstacles or walls. Finally, it is shown in Figure 14 that the safest trajectories were provided by the EFSP.

In the tests that were simulated by users, closest distances to the obstacles ($d_{min}$), risk factor ($r_f$), extended risk factor ($er_f$), user speed reference ($V_{user}$), and linear speed of the wheelchair ($V_{out}$) data were
collected for a numerical performance comparison. A specific safety metric for the comparison of trajectories is
defined in Eq. (12). This metric is based on the minimum distance to obstacles. Similar metrics are defined in [11,23,29].
\[
f(t) = \begin{cases} 
\frac{1}{d_{min}} - \frac{1}{d_0}, & \text{for } d_{min} < d_0 \\
0, & \text{for } d_{min} \geq d_0 
\end{cases}
\]  
(12)

where \(d_{min}\) is the closest distance between the vehicle and obstacles, and the given scalar, \(d_0\), denotes the
distance to an obstacle that poses no danger for collision during execution. The \(p\)th norm of a function is
defined as illustrated in Eq. (13):
\[
\|f\|_p = \left( \int |f(t)|^p \, dt \right)^{1/p}
\]  
(13)

In this paper, the 2nd norm (\(p = 2\)) of the safety metric is calculated and divided by the total travel time as a
performance criterion.
The numerical comparison of each approach in terms of collision rates, average safety metric, average wheelchair speed, average user speed, average risk factor, average minimum distance to obstacles, and average travel time is shown in Table 3.

Table 3. Summary of simulation results (WSP: without speed planner; VFFSP: virtual force field speed planner; FSP: fuzzy speed planner; EFSP: extended fuzzy speed planner).

<table>
<thead>
<tr>
<th>Method</th>
<th>Average safety metric</th>
<th>Total number of collisions</th>
<th>User’s average speed reference (m/s)</th>
<th>Wheelchair’s average speed (m/s)</th>
<th>Average Risk Factor</th>
<th>Average minimum distance to obstacles (m)</th>
<th>Average Travel time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSP</td>
<td>2.2292</td>
<td>3</td>
<td>1.7537</td>
<td>1.7462</td>
<td>0.1854</td>
<td>0.7163</td>
<td>30.06</td>
</tr>
<tr>
<td>VFFSP</td>
<td>1.9387</td>
<td>1</td>
<td>1.7606</td>
<td>1.6360</td>
<td>0.1697</td>
<td>0.7944</td>
<td>31.87</td>
</tr>
<tr>
<td>FSP</td>
<td>1.9015</td>
<td>0</td>
<td>1.7420</td>
<td>1.5109</td>
<td>0.1633</td>
<td>0.8216</td>
<td>34.39</td>
</tr>
<tr>
<td>EFSP</td>
<td>0.7863</td>
<td>0</td>
<td>1.7502</td>
<td>1.3873</td>
<td>0.1458</td>
<td>0.9328</td>
<td>37.12</td>
</tr>
</tbody>
</table>

As shown in Table 3, the speed planning method that was designed using FSP is 14.7% safer than WSP and 2% safer than VFFSP. Moreover, the newly designed extended fuzzy speed planner EFSP results in 65% safer trajectories than WSP, 59.4% safer trajectories than VFFSP, and 58.6% safer trajectories than FSP. Both FSP and EFSP had no collisions, while there were 3 collisions in WSP and 1 collision in VFFSP. As additional information, the collision trajectories were not integrated into safety metric calculations.

According to the results, the effect of using a speed planner is critically important to prevent possible collisions. Simulation results show that the average linear speed of the wheelchair obtained by EFSP is 20.6% slower than WSP. This is a reasonable rate, since we prevent collisions with speed planner support. In the tests performed with EFSP, the linear speed of the wheelchair was found to be 8.2% slower than that of FSP and 15.2% slower than that of VFFSP. Comparing these planning methods, the rate of decrease of the safety metric in EFSP is higher than the rate of decrease of the average linear speed. The average minimum distance value of each method illustrates the same behavior from a different perspective. Even though the extended fuzzy speed planner decreases the linear speed and increases the average travel time as expected, safety has been increased significantly, as shown in Table 3.

4. Hardware implementation and real experimental test

4.1. Experimental platform and test area

In the real environment tests, the extended fuzzy speed planning method was used as a speed planner while FGM-I was used for obstacle avoidance. The LIDAR in front of the wheelchair was used for obstacle detection, while the goal angle and linear speed information were obtained from the user with the orientation sensor located at the user’s head (Figure 10). More information about this platform’s head equipment can be found in [30]. A computer with 16 GB of RAM and 3.5 GHz processor speed was used for real-time computations. The experimental test area is shown in Figure 15, with the system and obstacles.

As seen in Figure 15, various forms of obstacles were scattered around for the real experimental tests. In real tests, \(d_{\text{min}}\), \(\Phi_{\text{obs}}\), \(r_f\), \(er_f\), \(V_{\text{user}}\), \(V_{\text{fuzzy}}\) and \(V_{\text{out}}\) are collected for analysis. Two different trajectories were tracked in two scenarios by the user.
4.2. Real experimental tests

Images that were recorded during the first test are given in Figure 16, while the related experimental data are shown in Figure 17. Figures 18 and 19 illustrate the same type of data for Test 2.

Figure 15. Experimental test area with wheelchair and obstacles.

Figure 16. Recorded images during Test-1.

Figure 17. Real experimental data of Test-1.

As can be seen from the results, $rf$ increases when $d_{min}$ and $\Phi_{obs}$ decrease. One can better see the difference between $rf$ and $erf$ when the user’s speed reference value is relatively high. Finally, the results validate that the effect of $V_{fuzzy}$ is greater than that of $V_{user}$ in regions where the $erf$ is high. The recorded video of Test 1 can be watched online at https://www.youtube.com/watch?v=OSlAuAsCB9I.

5. Conclusions and future work

In this paper, a new fuzzy logic-based semiautonomous speed planner was developed and applied to a real platform. We first improved a previously designed fuzzy planner and adapted it for semiautonomous systems.
Figure 18. Recorded images during Test-2.

Figure 19. Real experimental data of Test-2.

Instead of planning with only the classical risk factor, which is calculated using environmental factors only, we made a new risk definition, $erf$, in this paper. This new definition considers not only the distances and positions of the obstacles but also the user’s speed reference. Additionally, in this study we adapted both the classical fuzzy speed planner and extended fuzzy speed planner for semiautonomous systems. The comparative simulation results show that speed planning support prevents possible collisions and provides safer trajectories. Moreover, the extended risk factor definition increases safety considerably, without a significant reduction in speed. The real tests show that the developed approach works reliably in real time.

It should be noted that the extended planner can be used as a fully autonomous solution. Such a study can be carried out in the future in order to analyze the effect on fully autonomous systems.

Acknowledgment

This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under Project No. 215E140.
References


