



Social Hybrid Reciprocal Velocity Obstacle-Based Socially Aware Mobile Robot Navigation Framework

Duy Thao Nguyen, Van Bay Hoang, Trong Nghia Le, Tran Hiep Nguyen,
and Xuan Tung Truong^(✉)

Le Quy Don Technical University, Hanoi 6000000, Vietnam

Abstract. In this paper, we propose a social hybrid reciprocal velocity obstacle (SHRVO)-based navigation framework which allows a mobile robot to avoid safely and socially humans or groups of humans in dynamic social environments. The integration of social behaviors, including passing to the right of humans or human groups and selecting the left to overtake, into the original hybrid reciprocal velocity obstacle algorithm is the key idea of our proposed framework. The developed framework is evaluated by conducting a sequence of experiments in simulation environment. The experimental results have demonstrated the success of the proposed framework in safely navigating a mobile robot in dynamic social environment while still adhering to the social manners.

Keywords: Socially aware robot navigation · Cooperative navigation · Motion planning · Collision avoidance

1 Introduction

Recently, many socially aware navigation frameworks [7, 13] and [1] have been presented to address the problem of navigating mobile robots autonomously in public environments such as museums, airports, commercial centers and urban areas. Depending on how to integrate the human information and social restraints into the navigation systems, these current humanly aware systems can be relatively separated into: (i) social costmap-based approaches and (ii) motion planning system-based approaches. While the methods in the first category incorporate the human data and social behaviors to build the costmap function, the techniques in the second category directly integrate the social restriction into the motion planning systems.

Several navigation systems for mobile robots [6, 9, 10] and [18] have been introduced based on the social costmap techniques in recent times. In these systems, the 2-D Gaussian and linear techniques are applied to combine the human information and social constraints with the aim of building the social costmap. After that, the path planning algorithms [4, 16] and dynamic window

approach [3] are utilized to generate a feasible trajectory and a motion command to drive the robots to their targets. When these techniques are installed, the robots not only can avoid safely the surrounding humans, but also behave socially such as ensuring no violation of private space and social interaction area and selecting an appropriate side to overtake a person. However, when operating in crowded and dynamic environments [9] and [8], the computation process of these methods to determine a feasible path for mobile robots is very complicated and takes a lot of time.

In terms of approaches based on motion planning systems, the recent proposed navigation frameworks [2, 14, 17] and [12] have been developed by incorporating the information of humans into the conventional motion planning algorithms, for instance social force model [5] and velocity obstacles [15]. The computation time to generate the optimal path for mobile robots has been significantly improved and these techniques also have been verified in real-world environments. However, the social constraints such as taking the right side to pass and selecting the left to overtake are not considered in these systems.

In this paper, a human-like navigation system for mobile robots is proposed by incorporating the social rules into the original hybrid reciprocal velocity obstacle (HRVO) approach [15], which is a real-time algorithm considering the cooperation between robots operating in the shared environments. Particularly, we incorporate the social behaviors including selecting the right side to pass a human or a human group and taking the left side when overtaking, into the HRVO model, and propose a social hybrid reciprocal velocity obstacle (SHRVO) model. When equipped with the proposed SHRVO model, the mobile robots are capable of behaving humanly when avoiding humans and human groups, and navigating safely towards their targets.

The remainder of the paper is arranged as follows. Section 2 introduces the proposed social hybrid reciprocal velocity obstacle model. The simulation experiments are presented in Sect. 3. The conclusion of the paper is provided in Sect. 4.

2 Proposed Framework

2.1 Problem Description

Considering a socially dynamic environment which consists of a mobile robot and N surrounding humans. The mobile robot has a fixed radius r_A , a current position p_A , an actual velocity \mathbf{v}_A , a goal situated at p_A^{goal} and a preferred speed p_A^{pref} . The goal can be directly selected as a fixed position in the plane or be the consequence of some algorithms, for instance global planning or scheduling algorithms. The preferred speed is identified as the speed taken by the robot when there are no humans or obstacles in its path. Similar to choosing the goal of the mobile robot, this speed can be pre-selected or be the output of some external methods. In addition, humans presenting around the robot can be denoted as $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N\}$, where \mathbf{p}_i is the i^{th} person. Each human also has a constant radius r_i , a current position p_i , an actual velocity \mathbf{v}_i , a goal situated at p_i^{goal} and a preferred speed p_i^{pref} .

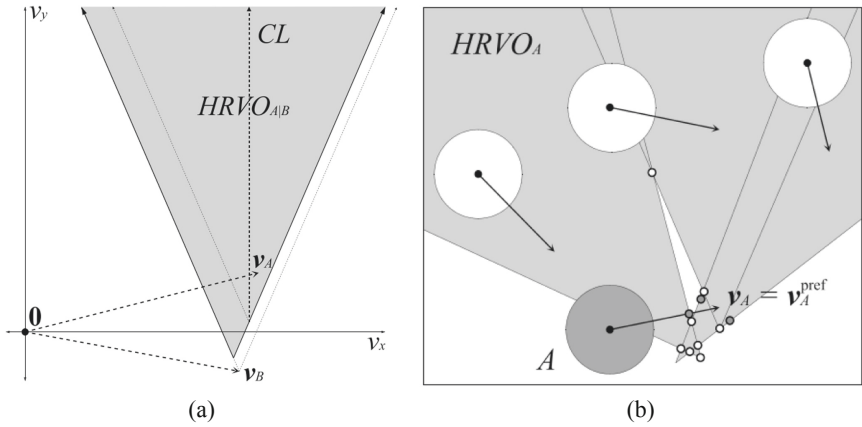


Fig. 1. (a) The configuration of HRVO. (b) The construction of the combined hybrid reciprocal velocity obstacle for the dark robot.

The objective of the mobile robot is to independently and concurrently compute a new velocity at each time step to generate an optimal trajectory towards its desired destination while not only ensuring the collision-free and oscillation-free with any human or obstacle in the environment but also following the social rules such as overtaking to the left of a human or a human group and passing them on the right side.

2.2 Hybrid Reciprocal Velocity Obstacle Model

The hybrid reciprocal velocity obstacle concept [15] is a real-time navigation algorithm for multiple mobile robots or virtual agents in which the reciprocity between robots is specially considered. The hybrid reciprocal velocity obstacle was developed to resolve one of the particular limitations of the reciprocal velocity obstacle formulation which is known as “reciprocal dance”, and this algorithm has been successfully applied to mobile robots in an real-world environment. A brief review of the concept of the hybrid reciprocal velocity obstacle will be described in this section.

To counter the situation of “reciprocal dances”, the hybrid reciprocal velocity obstacle combines the velocity obstacle and the reciprocal velocity obstacle to discourage robots from passing each other on different sides. For two robots A and B, if \mathbf{v}_A is to the right of the centerline of the reciprocal velocity obstacle for robot A induced by robot B ($\mathbf{RVO}_{A|B}$), with the aim of encouraging robot A to pass on the right side of robot B, the reciprocal velocity obstacle will be expanded to the left side by replacing the left edge of $\mathbf{RVO}_{A|B}$ with the left edge of the velocity obstacle $\mathbf{VO}_{A|B}$. The intersection point between the right side of $\mathbf{RVO}_{A|B}$ and the left side of $\mathbf{VO}_{A|B}$ will be the apex of the hybrid reciprocal velocity obstacle. We exchange right and left and mirror the procedure if \mathbf{v}_A is to the left of the centerline. The geometric interpretation of constructing a

hybrid reciprocal velocity obstacle $\mathbf{HRVO}_{A|B}$ for robot A induced by robot B is shown in Fig. 1(a).

In stead of having only two robots A and B, we assume that there is a set of robots \mathbf{A} operating together in an environment with a set of obstacles \mathbf{O} . Each robot A_i in \mathbf{A} has a current position p_{A_i} , velocity \mathbf{v}_{A_i} , radius r_{A_i} and goal position $p_{A_i}^{goal}$ that can be recognized or measured by other robots. Each robot A_i also has a preferred velocity $\mathbf{v}_{A_i}^{pref}$ toward its goal (velocity that a robot takes when there are no other robots or obstacles in its path) which is defined as follow:

$$\mathbf{v}_{A_i}^{pref} = v_{A_i}^{pref} \frac{\mathbf{P}_{A_i} - \mathbf{P}_{A_i}^{goal}}{\|\mathbf{P}_{A_i} - \mathbf{P}_{A_i}^{goal}\|_2} \quad (1)$$

Each robot A_i will independently execute a continuous cycle including: (i) Sense positions and velocities of nearby robots; (ii) Select optimal new velocity outside union of HRVO's; (iii) Compute control input from new velocity; (iv) and Apply control input to robot's actuators until reaching its goal position.

Based on information from the sensing phase, the hybrid reciprocal velocity obstacles \mathbf{HRVO} induced by each neighboring robot in \mathbf{A} and the velocity obstacles \mathbf{VO} generated by each obstacle in \mathbf{O} for robot A_i will be constructed and then unified to form the combined hybrid reciprocal velocity obstacle.

$$\mathbf{HRVO}_{A_i} = \bigcup_{A_j \in \mathbf{A}} \mathbf{HRVO}_{A_i|A_j} \cup \bigcup_{O_j \in \mathbf{O}} \mathbf{VO}_{A_i|O_j} \quad (2)$$

Because the data that are obtained from sensors and used for calculating the hybrid reciprocal velocity obstacle contain uncertainty, the combined uncertainty-adjusted hybrid reciprocal velocity obstacle will be constructed to ensure the correct functioning of HRVO approach by applying a Kalman filter.

For collision avoidance, the robot A_i should select a new velocity $\mathbf{v}_{A_i}^{new}$ that is closest to its preferred velocity $\mathbf{v}_{A_i}^{pref}$ and outside the combined uncertainty-adjusted hybrid reciprocal velocity obstacle:

$$\mathbf{v}_{A_i}^{new} = \arg \min_{\mathbf{v} \notin \mathbf{HRVO}_{A_i}} \|\mathbf{v} - \mathbf{v}_{A_i}^{pref}\|_2 \quad (3)$$

The ClearPath efficient geometric algorithm is used to compute this velocity. Due to the kinematic restrictions of the robot, the velocity $\mathbf{v}_{A_i}^{new}$ will be transformed into a control input (wheel speeds v_l and v_r) to allow the robot quickly reach the $\mathbf{v}_{A_i}^{new}$. This cycle will be continued until the robot reaches its goal position. This is illustrated in Fig. 1(b).

Algorithm 1 describes overall approach for navigating multiple mobile robots by applying the concept of HRVO, which is divided into three primary steps: (i) *perception* (Lines 3–4 of the Algorithm 1), *construction* (Lines 5–11 of the Algorithm 1), and (iii) *selection* (Lines 12–13 of the Algorithm 1). The input of the Algorithm 1 is the set of robots \mathbf{A} and set of obstacles \mathbf{O} , and the output is the control input wheel speeds $\mathbf{v}_r, \mathbf{v}_l$ of the mobile robot. *The first step is perception* and in which each robot in \mathbf{A} acquires its own position and velocity, and those of

Algorithm 1: Hybrid reciprocal velocity obstacle algorithm

```

input : Set of robots  $\mathbf{A}$ , set of obstacles  $\mathbf{O}$ 
output: Motion control command  $\mathbf{v}_r, \mathbf{v}_l$ 
1 begin
2   for all  $\mathbf{A}_i \in \mathbf{A}$  do
3      $\mathbf{S} \leftarrow \text{Sensing}(\mathbf{S});$ 
4      $\mathbf{G} \leftarrow \text{Gaussiandistribution}(\mathbf{S});$ 
5      $\mathbf{VO} \leftarrow \text{ConstructVO}(\mathbf{A}, \mathbf{O}, \mathbf{S});$ 
6      $\mathbf{RVO} \leftarrow \text{ConstructRVO}(\mathbf{VO}, \mathbf{S});$ 
7      $\mathbf{LC} \leftarrow \text{LocateCenterline}(\mathbf{RVO});$ 
8      $\mathbf{CH} \leftarrow \text{ConstructHRVO}(\mathbf{VO}, \mathbf{RVO});$ 
9      $\mathbf{AH} \leftarrow \text{AdjustHRVO}(\mathbf{G}, \mathbf{CH});$ 
10     $\mathbf{AVO} \leftarrow \text{AdjustVO}(\mathbf{G}, \mathbf{VO});$ 
11     $\mathbf{CAH} \leftarrow \text{CombineAdjustHRVO}(\mathbf{AH}, \mathbf{AVO});$ 
12     $\mathbf{PV} \leftarrow \text{PrefVelocity}(\mathbf{A}, \mathbf{S});$ 
13     $\mathbf{NV} \leftarrow \text{NewVelocity}(\mathbf{CAH}, \mathbf{PV});$ 
14  end for
15  Return  $\mathbf{v}_r, \mathbf{v}_l;$ 

```

the surrounding robots and obstacles in \mathbf{O} . Because sensors contain uncertainty, a Kalman filter is used to obtain the accurate estimation of the positions and velocities of the robots and obstacles by providing a Gaussian distribution of these information. *In the second step*, the velocity obstacles \mathbf{VO} and reciprocal velocity obstacles \mathbf{RVO} for each robot in \mathbf{A} induced by other surrounding robots in \mathbf{A} and velocity obstacles in \mathbf{O} are constructed. Next, depending on the relative position between the centerline of \mathbf{RVO} and the velocity of the robot, one of two edges of \mathbf{RVO} will be replaced with a corresponding edge of \mathbf{VO} to construct the hybrid reciprocal velocity obstacle \mathbf{HRVO} . After that, the \mathbf{HRVO} will be expanded to the uncertainty-adjusted hybrid reciprocal velocity obstacle \mathbf{HRVO}^* by using the Gaussian distribution to reduce the uncertainty in the perception step. The velocity obstacle $\mathbf{VO}_{A|O}$ for each obstacle in \mathbf{O} in the environment are also transferred to uncertainty-adjusted velocity obstacle \mathbf{VO}^* . The last part of this step will be the construction of the combined uncertainty-adjusted hybrid reciprocal velocity obstacle for each robot in \mathbf{A} . *In the final selection step*, a new velocity for each robot will be computed by using the Clearpath algorithm before converting to the speeds of the wheels which are the control input of each robot.

2.3 Proposed Social Hybrid Reciprocal Velocity Obstacle Algorithm

In HRVO concept, for two robots A and B, whether robot A will choose a new velocity to the right or left of $\mathbf{RVO}_{A|B}$ depending on the relative position between \mathbf{v}_A and the centerline of $\mathbf{RVO}_{A|B}$. In our social HRVO concept, in addition to relative position, we also consider the distance between the current velocity of robot A and the centerline of $\mathbf{RVO}_{A|B}$ when choosing the new

Algorithm 2: Social hybrid reciprocal velocity obstacle algorithm

```

input : Set of robots  $\mathbf{A}$ , set of obstacles  $\mathbf{O}$ 
output: Motion control command  $\mathbf{v}_r, \mathbf{v}_l$ 
1 begin
2   for all  $\mathbf{A}_i \in \mathbf{A}$  do ;
3      $\mathbf{S} \leftarrow \text{Sensing}(\mathbf{S})$ ;
4      $\mathbf{G} \leftarrow \text{Gaussiandistribution}(\mathbf{S})$ ;
5      $\mathbf{T} \leftarrow \text{setupThreshold}()$ ;
6      $\mathbf{VO} \leftarrow \text{ConstructVO}(\mathbf{A}, \mathbf{O}, \mathbf{S})$ ;
7      $\mathbf{AVO} \leftarrow \text{AdjustVO}(\mathbf{G}, \mathbf{VO})$ ;
8      $\mathbf{RVO} \leftarrow \text{ConstructRVO}(\mathbf{VO}, \mathbf{S})$ ;
9      $\mathbf{LC} \leftarrow \text{LocateCenterline}(\mathbf{RVO})$ ;
10    if distance ( $\mathbf{v}_{A_i}$ , centerline) < threshold;
11       $\mathbf{CHS} \leftarrow \text{ConstructSHRVO}(\mathbf{VO}, \mathbf{RVO}, \mathbf{LC}, \mathbf{T})$ ;
12       $\mathbf{AHS} \leftarrow \text{AdjustSHRVO}(\mathbf{G}, \mathbf{CHS})$ ;
13       $\mathbf{CASH} \leftarrow \text{CombineAdjustSHRVO}(\mathbf{AHS}, \mathbf{AVO})$ ;
14    else
15       $\mathbf{CH} \leftarrow \text{ConstructHRVO}(\mathbf{VO}, \mathbf{RVO})$ ;
16       $\mathbf{AH} \leftarrow \text{AdjustHRVO}(\mathbf{G}, \mathbf{CH})$ ;
17       $\mathbf{CAH} \leftarrow \text{CombineAdjustHRVO}(\mathbf{AH}, \mathbf{AVO})$ ;
18       $\mathbf{PV} \leftarrow \text{PrefVelocity}(\mathbf{A}, \mathbf{S})$ ;
19       $\mathbf{NV} \leftarrow \text{NewVelocity}(\mathbf{CAH}, \mathbf{PV})$ ;
20    end for
21    Return  $\mathbf{v}_r, \mathbf{v}_l$ ;

```

velocity for robot A. We install a number that plays a role as a threshold for this selection. If the distance between the velocity and centerline of $\mathbf{RVO}_{A|B}$ is larger than the threshold, robot will follow the HRVO concept for choosing new velocity (Lines 14–17 of the Algorithm 2). If the distance is smaller than the threshold and the velocity is not to the side as we wish, the centerline of $\mathbf{RVO}_{A|B}$ will be moved an equal distance to the threshold and to the side that we do not wish the robot A to pass (Lines 10–13 of the Algorithm 2). This transfer ensures that \mathbf{v}_A is to the desired side of the centerline of $\mathbf{RVO}_{A|B}$. By the symmetry, this is totally similar to robot B. Consequently, robots can pass others on the side that is in accordance with social rules. Because of adding the social rules to the HRVO concept, we call the result a social hybrid reciprocal velocity obstacle, and denote it $\mathbf{SHRVO}_{A|B}$. The method to construct the social hybrid reciprocal velocity obstacle is described specifically in Algorithm 3. The input of Algorithm 3 includes the radius, position, current velocity and preferred velocity of all robots in the environment, while the output will be the construction of social hybrid reciprocal velocity obstacle consisting of apex, left side and right side. The Algorithm 3 first calculates the parameters that will be used to create the \mathbf{VO} , \mathbf{RVO} and declares a threshold as a condition for constructing the \mathbf{SHRVO} (Lines 4–9 of the Algorithm 3). If the velocity of robot is to the right (or left) of the centerline of the \mathbf{RVO} but the distance between them is less

Algorithm 3: Construct SHRVO

```

input : Set of robots  $\mathbf{A}$  with: robot position  $\mathbf{p}_{A_i}$ , robot velocity  $\mathbf{v}_{A_i}$ , robot
         preferred velocity  $\mathbf{p}_{A_i}^{pre}$ , robot radius  $r_{A_i}$ 
output: SHRVO (apex, side1,side2)
1 begin
2   for all  $A_i \in \mathbf{A}$  do
3     for all  $A_j \in \mathbf{A} (j \neq i)$  do
4       constant bias: threshold;
5       Angle =  $\text{atan}(p_{A_j} - p_{A_i})$ ;
6       openingAngle =  $\frac{\text{asin}(r_{A_j} - r_{A_i})}{\text{abs}(p_{A_j} - p_{A_i})}$ ;
7       d =  $2 * \sin(\text{openingAngle}) * \cos(\text{openingAngle})$ ;
8       VOSide1 =  $(\cos(\text{Angle} - \text{openingAngle}), \sin(\text{Angle} - \text{openingAngle}))$ ;
9       VOSide2 =  $(\cos(\text{Angle} + \text{openingAngle}), \sin(\text{Angle} + \text{openingAngle}))$ ;
10      if  $\det(p_{A_j} - p_{A_i}, p_{A_i}^{pre} - p_{A_j}^{pre}) > 0$ ;
11        if  $\text{DispPointToSegment}(p_{A_j} - p_{A_i}, p_{A_i}^{pre} - p_{A_j}^{pre}) < \text{bias}$ ;
12          VBnew =  $\text{ClosePointOnSegment}(v_{A_j}, (v_{A_j} - \text{bias}))$ ;
13          interPoint =
14             $\text{InterVectors}(\text{VBnew}, \text{VOSide2}, 0.5 * (v_{A_i} + v_{A_j}), \text{VOSide1})$ ;
15            HRVOapex =
16               $\text{interPoint} - \text{norm}(p_{A_j} - p_{A_i}) * \frac{\text{unOffset} * \text{abs}(p_{A_j} - p_{A_i})}{r_{A_j} + r_{A_i}}$ ;
17            else
18              s =  $0.5 * \det(v_{A_j} - v_{A_i}, \text{VOSide2})$ ;
19              HRVOapex =
20                 $v_{A_j} + s * \text{VOSide1} - \text{norm}(p_{A_j} - p_{A_i}) * \frac{\text{unOffset} * \text{abs}(p_{A_j} - p_{A_i})}{r_{A_j} + r_{A_i}}$ ;
21            else
22              s =  $0.5 * \det(v_{A_j} - v_{A_i}, \text{VOSide1})$ ;
23              HRVOapex =
24                 $v_{A_j} + s * \text{VOSide2} - \text{norm}(p_{A_j} - p_{A_i}) * \frac{\text{unOffset} * \text{abs}(p_{A_j} - p_{A_i})}{r_{A_j} + r_{A_i}}$ ;
25            end for;
26          end for;
27        Return SHRVO(apex, side1,side2);

```

than the threshold, the **SHRVO** will be generated to make the robot act like human in society (Lines 10–14 of the Algorithm 3). If this distance is larger than the threshold, the robot will follow the concept of **HRVO** (Lines 16–20 of the Algorithm 3).

The geometric interpretation of **R-SHRVO** for the right rule is illustrated in Fig. 2(a). Because the distance between \mathbf{v}_A and CL is smaller than the threshold b , robot A will choose a new velocity to the right of $\mathbf{RVO}_{A|B}$ even though \mathbf{v}_A is to the left of the centerline of $\mathbf{RVO}_{A|B}$. To encourage this, a right social HRVO

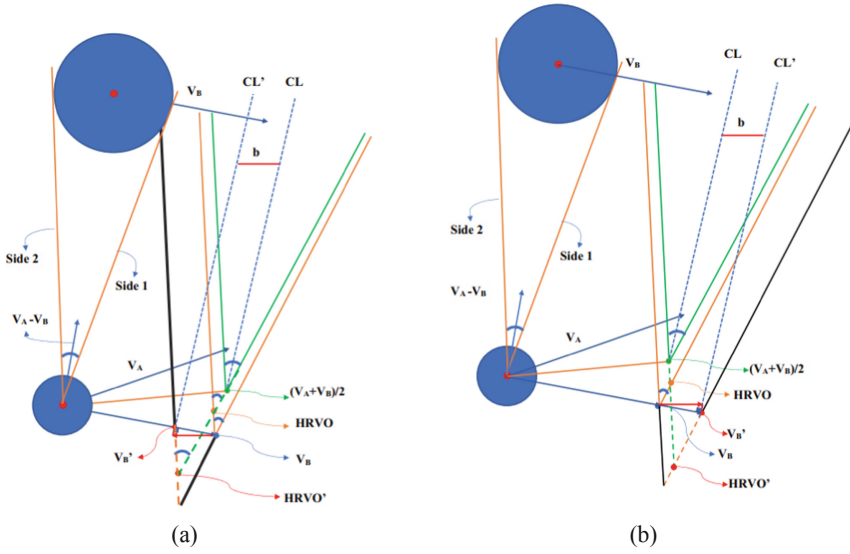


Fig. 2. The first figure is the geometric interpretation of R-SHRVO for the right rule and the second is the geometric interpretation of R-SHRVO for the left rule

is created by enlarging the reciprocal velocity obstacle to the left side. The steps are as follows:

Step 1 - Create a new velocity obstacle (VO'): From the apex of VO (v_B), draw a vector v_b that is parallel to the Ox axis (origin of the coordinated axis is at the center of robot A), toward the left and has a length of threshold b .

$$v_b = v_B - (b, 0) \tag{4}$$

Project v_b onto velocity v_B , we get point B' . This is the new apex of $VO'_{A|B}$. And from this apex, draw two edges that are parallel to the edges of $VO_{A|B}$ to form a new velocity obstacle $VO'_{A|B}$.

*Step 2 - Create a **SHRVO** for right rule (R -SHRVO):* The apex of **R-SHRVO** is the intersection between the right side of $RVO_{A|B}$ and the left side of $VO'_{A|B}$.

$$\text{apex} = \text{right } RVO_{A|B} \cap \text{left } VO'_{A|B} \tag{5}$$

As a result, the left boundary of **SHRVO** for right rule will be the left side of $VO'_{A|B}$, and the right boundary will be the right side of $RVO_{A|B}$.

For the left rule, the procedure is nearly the same as described above, just exchanging right and left. The $RVO_{A|B}$ at that time will be expanded to the right to encourage robot A to take the left side of robot B to pass. Figure 2(b) will illustrate the steps to construct the **SHRVO** for left rule.

The apex of **L-SHRVO** is the intersection between the left side of $\mathbf{RVO}_{A|B}$ and the right side of $\mathbf{VO}'_{A|B}$. Two boundaries of **SHRVO** for left rule will be the right side of $\mathbf{VO}'_{A|B}$, and the left side of $\mathbf{RVO}_{A|B}$.

$$\mathbf{apex} = \text{left } \mathbf{RVO}_{A|B} \cap \text{right } \mathbf{VO}'_{A|B} \quad (6)$$

3 Experiments

The effectiveness of the proposed social HRVO model is verified by implementing and testing in a simulation environment. The C/C++ programming language is applied to implement the proposed system, while the Robot Operating System (ROS) [11] is utilized to develop the entire navigation framework. The original HRVO package¹ was inherited and adjusted to develop the proposed social HRVO model.

3.1 Simulation Setup

We create scenarios, as shown in Figs. 3, 4, 5 and 6, in which the mobile robot need to cooperate with the behaviors of other humans in the environments to reach its goal. In addition to guarantee the collision-free, the robot is required to follow the social rules such as passing on the right side or left side of other humans. With the aim of demonstrating the performance of the proposed social HRVO model, we compare it with the original hybrid reciprocal velocity obstacle model HRVO [15]. Furthermore, the proposed socially aware navigation framework is validated by adopting the human comfortable safety indices (HCSI) introduced by Truong et al. [19], including *social individual index (SII)*, *social group index (SGI)*. The HCSI indices are used to estimate the human safety and comfort and socially acceptable behaviors of the mobile robot. Specifically, the SII value is applied to measure the physical safety and psychological safety of each individual, while the SGI is used to measure the comfortable safety of the human group.

3.2 Simulation Results

Videos with our simulation results can be found at the links^{2,3,4} and⁵.

Experiment 1 – Avoiding a Person: We first examine the proposed SHRVO model in a simulation environment with a person. The mobile robot is expected to move from left to right without any collision with a moving person. The simulation results are shown in Fig. 3. The first and third rows show the trajectory

¹ <https://gamma.cs.unc.edu/HRVO/>.

² <https://youtu.be/wTTiu03l5Mw>.

³ <https://youtu.be/6gm5oFtz0uo>.

⁴ <https://youtu.be/GL03vP92J1Y>.

⁵ <https://youtu.be/nPKkuFw4z0Y>.

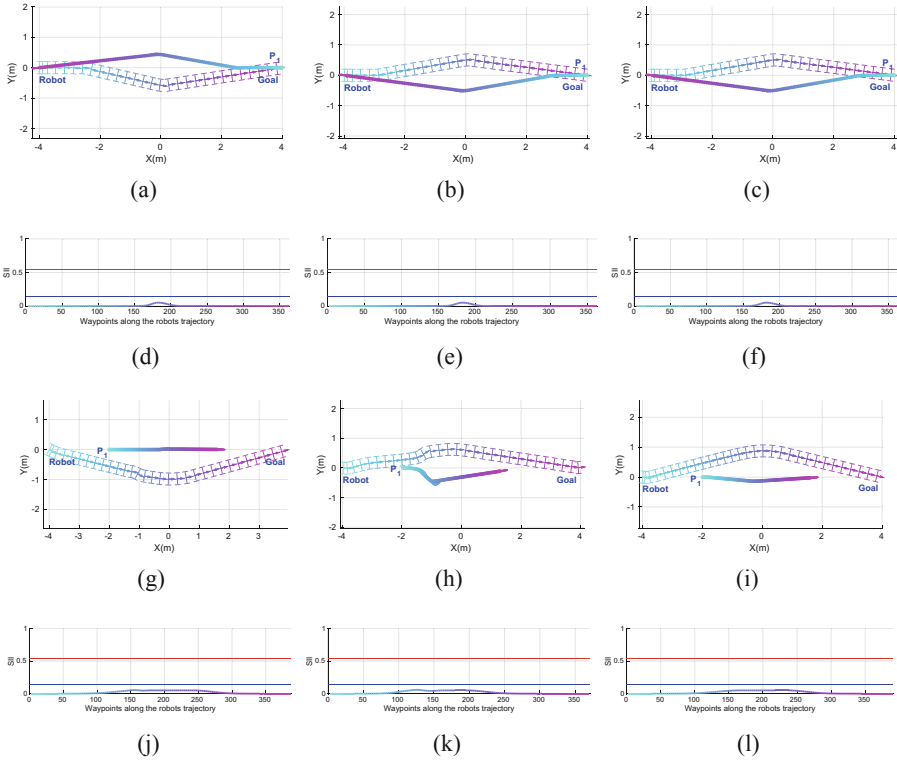


Fig. 3. The simulation results of cooperative avoidance of a robot when avoiding a person. The first and third rows, from left to right, show the trajectory of the robot and the human equipped with the R-SHRVO model, L-SHRVO model and the conventional HRVO model, respectively. The second and fourth rows illustrate the corresponding social individual index (SII).

of the robot and the person, and the second and fourth rows illustrate the SII value along the robot’s trajectory.

In Figs. 3(a) and 3(b), the mobile robot equipped with the proposed SHRVO technique successfully cooperates with the moving person to avoid each other on the side that we preconfigure in the algorithm. In contrast, the robot and the person select the left side to avoid if it is installed the conventional HRVO model in Fig. 3(c). Similarly, in Figs. 3(g) and 3(h), the robot successfully overtakes on the side following the rule which is installed in the SHRVO model, whereas left side is the selection of the robot and the person with the HRVO model as shown in Fig. 3(i).

Figures 3(d), 3(e), 3(f), 3(j), 3(k) and 3(l) show that, when equipped with the SHRVO model, the mobile robot not only successfully cooperates with the moving person to overtake or avoid but also still maintains a safety distance with the person.

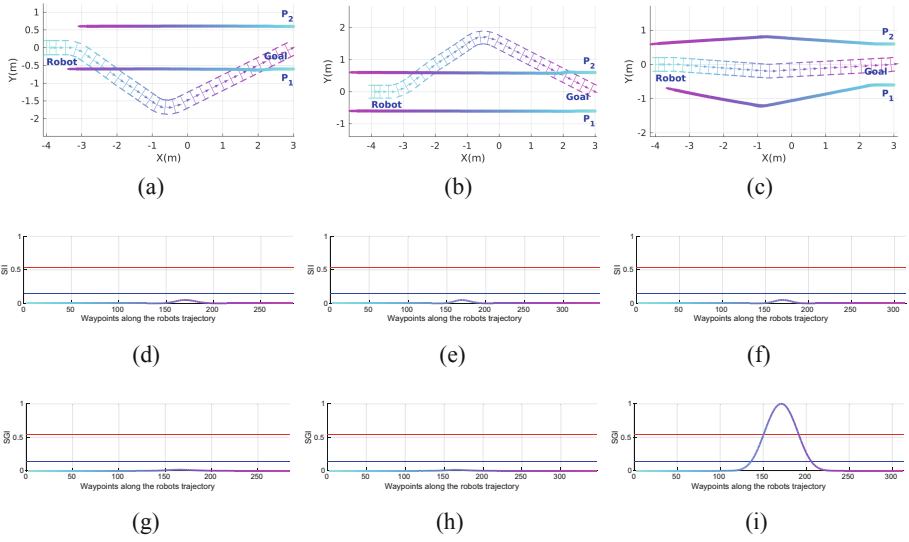


Fig. 4. The simulation results of cooperative avoidance of the robot and a group of two people in opposite directions. The first row, from left to right, shows the trajectory of the robot and the humans equipped with the R-HRVO model, L-HRVO model and the conventional HRVO model, respectively. The second and third rows illustrate the corresponding social individual index and social group index (SGI).

Experiment 2 – Avoiding a Group of Two People: In this experiment, we want to verify the avoiding behavior of the mobile robot when avoiding a group of two people moving towards the robot. The simulation results are shown in Fig. 4, in which the first row shows the trajectory of the robot when passing two people, while the second and last rows illustrate the SII and SGI values along the robot’s trajectory, respectively. In the first row of Fig. 4, by cooperating with the behaviors of two people, the mobile robot equipped with the proposed SHRVO algorithm, as presented in Figs. 4(a) and 4(b), or the HRVO algorithm, as presented in Fig. 4(c), is able to safely navigate to its goal.

Although the SII values in Fig. 4(d), Fig. 4(e) and Fig. 4(f) are similarly low indicating that the robot equipped with both SHRVO and HRVO models often maintains a safety distance with each person, the trajectory of HRVO model goes through the interaction space between two people. This is the reason for the much higher SGI value of HRVO model in Fig. 4(i) in comparison with the SGI values in Fig. 4(g) and Fig. 4(h) of SHRVO model. Therefore, in this experiment, the robot installed the SHRVO model offers a higher safety distance with the group of two people than the HRVO model.

Experiment 3 – Avoiding Dynamic People: The successful performance of the proposed SHRVO model is further clarified in several crowded environment

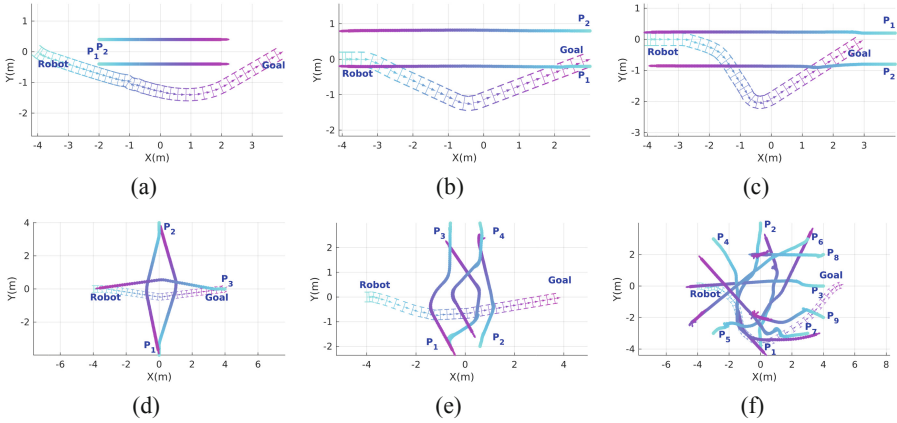


Fig. 5. The simulation results of cooperative avoidance of the robot equipped with R-SHRVO model when avoiding dynamic people. The first row shows the trajectory of the robot and two humans in different situations, while the second row, from left to right, illustrates the trajectory of the robot and three people, four people and a crowd of people, respectively.

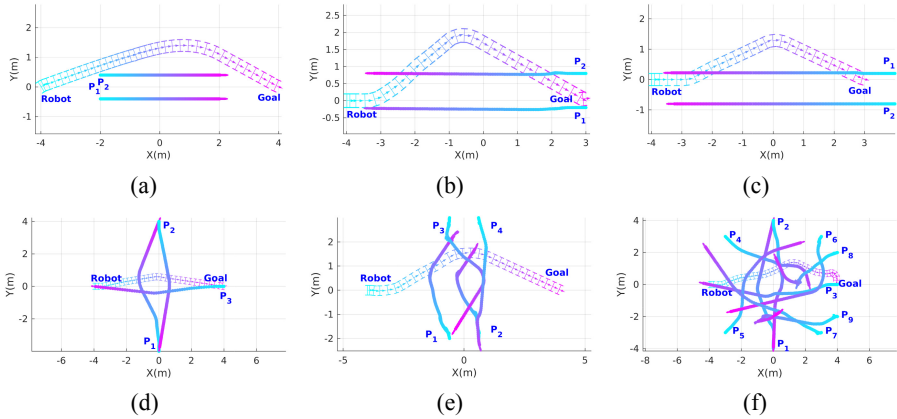


Fig. 6. The simulation results of cooperative avoidance of the robot equipped with L-SHRVO model when avoiding dynamic people. The first row shows the trajectory of the robot and two humans in different situations. Whereas, the second row, from left to right, illustrates the trajectory of the robot and three people, four people and a crowd of people, respectively.

scenarios, in which many people are moving toward the robot or in the same direction. The simulation results are shown in Fig. 5 and Fig. 6.

In Fig. 5, the simulation results show that the mobile robot equipped with the R-SHRVO algorithm successfully cooperates with the humans to guarantee the collision-free by selecting the right side to avoid each other. Especially, when

one of two people is to the right of the robot as shown in Fig. 5(c) or there are a crowd of people in the environment as shown in Fig. 5(f), the robot still chooses the right side to avoid the humans if it is installed the R-SHRVO model.

Similarly, Fig. 6 illustrates that the robot equipped with the L-SHRVO model also succeeds in navigating to its goal without any collision by selecting the left side to avoid the humans. The successful performance of the L-SHRVO model is demonstrated even if one of two people is to the left of the robot as shown in Fig. 6(b) or there are a crowd of people as shown Fig. 6(f).

In summary, the simulation results illustrate that, the robot equipped with the proposed SHRVO model is capable of cooperating with the humans in the crowded environment, and safely and socially navigating to the given target.

4 Conclusion

We have presented a social hybrid reciprocal velocity obstacle (SHRVO)-based navigation framework which ensures the free collision and social behavior for a mobile robot while navigating in dynamic social environments. In our proposed system, the social rules are integrated into the conventional HRVO algorithm to construct the SHRVO model. We have executed many simulation experiments to verify the proposed framework and the results demonstrate that our developed system can be able to navigate the mobile robot safely and humanly in dynamic social environments.

In the future, we will install the proposed SHRVO model on our mobile robot platform, and conduct experiments in real-world environments.

Acknowledgment. This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2018.10.

References

1. Cheng, J., Cheng, H., Meng, M.Q., Zhang, H.: Autonomous navigation by mobile robots in human environments: a survey. In: 2018 IEEE International Conference on Robotics and Biomimetics, pp. 1981–1986, December 2018
2. Ferrer, G., Sanfeliu, A.: Proactive kinodynamic planning using the extended social force model and human motion prediction in urban environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1730–1735, September 2014
3. Fox, D., Burgard, W., Thrun, S.: The dynamic window approach to collision avoidance. *IEEE Trans. Robot. Autom.* **4**(1), 23–33 (1997)
4. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **4**(2), 100–107 (1968)
5. Helbing, D., Molnár, P.: Social force model for pedestrian dynamics. *Phys. Rev. E* **51**, 4282–4286 (1995)
6. Kirby, R., Simmons, R., Forlizzi, J.: COMPANION: a constraint-optimizing method for person-acceptable navigation. In: Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication, pp. 607–612, September 2009

7. Kruse, T., Pandey, A.K., Alami, R., Kirsch, A.: Human-aware robot navigation: a survey. *Robot. Auton. Syst.* **61**, 1726–1743 (2013)
8. Kruse, T., Kirsch, A., Khambhaita, H., Alami, R.: Evaluating directional cost models in navigation. In: *Proceedings of the ACM/IEEE International Conference on Human-robot Interaction*, pp. 350–357 (2014)
9. Lu, D.V., Smart, W.D.: Towards more efficient navigation for robots and humans. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1707–1713. IEEE (2013)
10. Pandey, A.K., Alami, R.: A framework towards a socially aware mobile robot motion in human-centered dynamic environment. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5855–5860, October 2010
11. Quigley, M., et al.: ROS: an open-source robot operating system. In: *ICRA Workshop on Open Source Software*, vol. 32, pp. 151–170 (2009)
12. Repiso, E., Garrell, A., Sanfeliu, A.: Adaptive side-by-side social robot navigation to approach and interact with people. *Int. J. Soc. Robot.* **12**, 909–930 (2020). <https://doi.org/10.1007/s12369-019-00559-2>
13. Rios-Martinez, J., Spalanzani, A., Laugier, C.: From proxemics theory to socially-aware navigation: a survey. *Int. J. Soc. Robot.* **7**, 137–153 (2015). <https://doi.org/10.1007/s12369-014-0251-1>
14. Shiomi, M., Zanlungo, F., Hayashi, K., Kanda, T.: Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using a pedestrian model. *Int. J. Soc. Robot.* **6**(3), 443–455 (2014). <https://doi.org/10.1007/s12369-014-0238-y>
15. Snape, J., Van den Berg, J., Guy, S., Manocha, D.: The hybrid reciprocal velocity obstacle. *IEEE Trans. Robot.* **27**(4), 696–706 (2011)
16. Stentz, A.: The D* algorithm for real-time planning of optimal traverses. Technical report, CMU-RI-TR-94-37. The Robotics Institute, Carnegie-Mellon University (1994)
17. Truong, X.T., Ngo, T.D.: Toward socially aware robot navigation in dynamic and crowded environments: a proactive social motion model. *IEEE Trans. Autom. Sci. Eng.* **14**(4), 1743–1760 (2017)
18. Truong, X.T., Ngo, T.D.: Dynamic social zone based mobile robot navigation for human comfortable safety in social environments. *Int. J. Soc. Robot.* **8**(5), 663–684 (2016). <https://doi.org/10.1007/s12369-016-0352-0>
19. Truong, X.T., Ngo, T.D.: “To Approach Humans?”: a unified framework for approaching pose prediction and socially aware robot navigation. *IEEE Trans. Cogn. Dev. Syst.* **10**(3), 557–572 (2017)