International Journal of Social Robotics
https://doi.org/10.1007/s12369-020-00721-1

Social Cues in the Autonomous Navigation of Indoor Mobile Robots

Arun Kumar Reddy[1] · **Vaibhav Malviya**[1] · **Rahul Kala**[1]

Accepted: 26 October 2020

Abstract
Robots are now operating in workspaces occupied by humans and the robots often need to avoid people while travelling to their desired goal. Humans follow some navigation conventions while walking, called as social cues, like avoiding an oncoming person from the left. It is important for the robots to display the same social cues to be socially accepted by the humans. Current works in social robot motion planning are limited to maintenance of a socially compliant distance. This paper exhibits a socialistic behavior of a robot avoiding an oncoming human by the preferred side. This paper extends the social force model to incorporate the social cues by adding new social forces. The paper also extends the geometric approach to incorporate the social cues by selecting the geometric gap as per the social preference. Finally, we propose a novel hybrid approach combining the social potential field and geometric method, wherein the preferred gap for navigation adds a new social force to the robot. The proposed approach maintains the proactive nature of the geometric approach as well as retains the reactive nature of the social force model. The hybrid method maintains a larger clearance and generates safer trajectories as compared to the baseline social potential field and follow the gap methods. The experiments are done with the Pioneer LX robot using vision cameras and a lidar for navigation operating at the Centre of Intelligent Robotics at IIIT Allahabad.

Keywords Social robot motion planning · Social potential field · Geometric motion planning · Social robotics

1 Introduction

Due to advancements in robotic technologies in the recent years, there has been a lot of research in the area of social robotics [1, 2], where the robots work alongside humans. Such services include room service, luggage carrying services in areas such as malls, railway stations and airports [3–5]. These tasks are very complex because the human behaviors are unpredictable.

1.1 Social Cues

The humans usually follow a few subtle behaviors known as social cues for easy navigation in crowded environments. Social cues are usually non-verbal hints that can be used to guide interactions and help to clarify people's meanings and intentions. It is important that the robots exhibit social cues to be recognized by the humans and the social cues be like those used by the humans.

While navigating an environment with many humans around, we tend to maintain a reasonable distance from the others passing by, and the robots must do the same [6, 7]. It has been widely observed that the humans follow simple navigation rules to avoid colliding with each other such as passing on the left or overtaking on the right [8, 9]. These rules are derived from the traffic conventions using a left-handed convention, which may be reversed for a right-handed convention.

The first social cue introduced is passing on the left. If two humans walk towards each other head on, it is evident that both will drift leftwards to avoid each other. The specific behaviour is inspired by the traffic [9]. In countries where traffic is unorganized and lane discipline is not mandatory (like India), closely representing a mobile robotics problem, it is common for two vehicles to come head-on towards each other. However, both vehicles drift sideways (on their left if the traffic operates on the left and vice versa) in anticipation to let the other vehicle pass. The robot must similarly

✉ Vaibhav Malviya
rsi2016001@iiita.ac.in

Arun Kumar Reddy
irm2014005@iiita.ac.in

Rahul Kala
rkala@iiita.ac.in

[1] Centre of Intelligent Robotics, Indian Institute of Information Technology, Allahabad, Prayagraj, India

Published online: 12 November 2020

drift leftward on encountering a person directly ahead. The second social cue is allowing overtaking on the right. If a human is getting late and walking fast, with a slow-moving human ahead, an overtaking takes place. If the two humans are oriented in the same direction and the slower human is directly in front, the overtaking happens in a characteristic manner. The overtaking human drifts rightwards, while the human being overtaken drifts a little leftwards to allow the overtaking to take place. The other related social cue of overtaking on the right is not modelled as the robot rarely moves close to the human walking speed to have the robot overtake humans.

The human-in-the-loop is a very recent convention in robotics, wherein the robot is made to operate in the presence of humans [10, 11]. The field is difficult to study since the human behavior is difficult to model, meaning that the testing can only be done with real human subjects. The humans do display some general conventions of avoiding each other on the left, overtaking on the right, etc.; while these social conventions may not be the best as per any specific objective measure of path length, clearance, smoothness, etc. At the same time there are times where a display of the general conventions may not be possible due to contradictory goal, space constraints or any other reason.

The field of social robot motion planning [12, 13] enables making the robots operate in workspaces in the presence of humans, following socialistic behaviors while making the overall motion collision-free. The current approaches are limited to human–robot interaction, manipulation in cooperation with humans, extrapolating the human trajectory to avoid them, etc. There has been a very limited literature towards the navigation of robots in such contexts. The navigation literature is also limited to the maintenance of socialistic distances, which finds a lot of application in simulating crowd behaviors. The paper proposes to solve the problem of social robot motion planning. We first observe the generic behaviors that the humans involuntarily display while navigating in home or office environments. Thereafter, the robot navigation strategy is designed such that these social cues are anticipated as well as displayed by the robot.

1.2 A Brief of the Approach

There are two dominant techniques for the reactive navigation of the robot. The first is the social potential field [14, 15]. Here, every obstacle applies a repulsive force, while the robot experiences an attractive force towards the goal (or in the direction of motion). The method is social because of the socialistic modelling of the forces from the humans. We extend the social forces model by introducing additional forces to enforce the conventions to be followed by the robot.

The second dominant approach for the reactive navigation of the humans involves a geometric assessment of the obstacles [16]. Typically, the obstacles are assessed to get the gaps, within which the robot can place itself. Out of all the gaps, the best one is chosen, and the heading direction is kept to avoid the obstacles with a safe distance, while keeping generally towards the goal. We extend the geometric approach by introducing a two-stage gap qualification and gap-selection strategy, and then introduce social cues into the selection strategy in an intuitive manner.

Both potential field and geometric approaches have contrasting pros and cons which motivate the fusion of the two techniques making a hybrid reactive algorithm for the social navigation of the robot [17, 18]. We design a novel hybrid method by combining the force model with the geometric approach in a way that both their characteristics are retained and the social cues are displayed. We demonstrate that the hybrid method generates safer trajectories and maintains a larger clearance from the humans.

We perform both simulations and experiments on a Pioneer LX robot equipped with a lidar sensor and a camera in an environment with walking humans. By experiments we show that the proposed approach outperforms the baseline approaches by generating safer and socially compliant trajectories.

1.3 Main Contributions

The paper assumes a general social convention that the humans prefer to avoid each other from the left and allow aggressive people to overtake by not coming on their way. The novelty of the work is adapting the current reactive motion planning algorithms to include this social behavior into their working. It is one of the first approaches that directly translates the social human behavior into the existing motion planning methods from robotics literature. The main contributions are:

- Fusion of a geometric approach which finds the best social gap and a social potential approach for socialistic avoidance of the humans.
- Modelling a new socialistic force in the potential field approach, which is applied orthogonal to the vector to the human. The selection of leftward or rightward orthogonal direction vector is done based on the social preference.
- Using socialistic cues to select a gap for the navigation of the robot, to make the resultant approach socially compliant.
- The results are demonstrated under live settings using the Pioneer LX robot.
- Based on the experimental results, the hybrid approach performs better than the baseline and individual approaches.

2 Related Work

The existing literatures on decentralized, cooperative and socially compliant navigation can be broadly divided into either model-based approaches or learning-based approaches. Model based approaches are generally extensions of multi-agent collision avoidance algorithms by introducing additional parameters for social interactions. There can be different kinds of interactions such as human to robot interactions, robot to robot interactions etc. Learning based approaches on the other hand focus on finding the optimal policy that resembles human behaviors by using features obtained from the human trajectory data. It has been widely observed that humans follow certain conventions such as passing on the left and overtaking on the right according to the left-hand convention present in the UK, India etc. [19]. The same can be reversed such as passing on the right-hand convention present in the US, Japan, etc.

2.1 Geometry-Based Approaches

The classical planning algorithms can be used to generate a collision free path for the robot under the assumption of the hidden intents of the humans. It has however been observed that separating navigation into independent prediction and planning leads to the freezing robot problem. The robot in such cases fails to find a path as a major part of its environment is marked as untraversable. One of the most common approaches to the problem of collision avoidance with dynamic obstacles is to employ specific reactive rules in order to avoid collisions. Reciprocal velocity obstacle avoidance algorithms implicitly assume the behaviors of other agents to generate collision free trajectories [20–22]. Since these methods do not capture human behavior, they can sometimes end up generating dangerous trajectories. An important factor is to account for cooperative mechanisms between the robot and the humans to make the navigation easier. Cooperative mechanisms help to make navigation possible in some situations that could previously cause the freezing robot problem. An adaptive side-by-side navigation methodology has also been also introduced to guide the robot [23]. The approaches assume a well-known noise-free robot position and velocity, with a circular geometry. The approaches do not account for static obstacles of arbitrary shapes and rely on a deliberative technique to handle the static obstacles. This limits the performance when the reactive motion leads to a position contradictory to the deliberative plan.

The work of Pandey and Alami is conceptually closer to our work [8]. The authors modelled multiple rules along with a deliberative planning of the robot. Our approach is purely reactive in nature, which is more suited when multiple people walk around in the proximity, nullifying most of the initial deliberative planning. Pandey and Alami used a milestone-based approach. Our motive here is the fusion of the milestone-based approach with a social potential field approach, which by intuitions and experiments both produces better results than either of the social techniques. Unlike Pandey and Alami, we use the constants of potential field, geometric approach, and controllers to get the behaviors and only externally fuse a single behavior of social person avoidance. This method eliminates the problem of having an ensemble of rules that are hard to maintain. This method also avoids the problems of disagreement between rules when many of them are simultaneously valid. The method does not add selection parameters, while having an ability to demonstrate most of the behaviors.

Geometric approaches try to formulate a solution to the problem by finding the safest portion of the environment for the robot to traverse. The follow the gap method is a safety-first collision avoidance algorithm which works on the principle of finding the largest gap in its surroundings and moving through that gap [16]. The algorithm is very attractive as it requires a very little parameter tuning. The improved follow the gap method aims to fix one of the primary limitations of the follow the gap method which is that the extremely long trajectories often fail to reach the destination [24, 25]. One of the major limitations of the geometric approaches, which is an oscillatory behavior for very small changes in direction, remains. Vector Field Histogram (VFH) is another geometric approach and works based on calculating the obstacle density in the robot's surroundings [26]. The environment is divided into different sectors whose densities are found out and the sector with the least obstacle density is chosen to pass through by the robot. The VHF$^+$ method attempts to account for the size of the robot and the VHF* method aims to combine A* search algorithm in the search space to find the optimal path [27, 28]. Search algorithms are however time consuming and expensive to find acceptable paths in environments containing humans. The approach models all entities as static obstacles and does not account for social interactions.

2.2 Potential-Based Approaches

Artificial Potential Fields (APF) are a very popular class of techniques used for collision avoidance in static environments [14, 15, 29]. The potential field algorithms work on the idea that a virtual force field exists in the environment around the robot. The virtual force field is made up of a combination of attractive and repulsive forces applied by the surroundings. The goal/destination applies an attractive force on the robot while the obstacles apply a repulsive force on

the robot. These force models can be extended to account for social interactions by introducing additional parameters which come into play in the presence of humans [12, 30]. The social parameters are determined by running multiple real-life trials consisting of participants [31, 32]. The potential field techniques are computationally very efficient and therefore work very well in real time. The main concerns regarding these methods come from the fact that the parameters need to be tuned individually and can vary from scenario to scenario and the trajectories followed sometimes have an oscillatory behavior.

The overall problem in this paper is the same as Takahashi et al. [33]. However, while Takahashi et al. modelled the persons as obstacles with different force constants, the approach presented in our paper does a very different social force modelling. Here a force is applied orthogonal to the direction of the person. Our general observation in social potential field-based methods is that the robot gets too close to the person when the person comes ahead suddenly. Therefore, we model an additional orthogonal force that quickly de-aligns the robot when the person and robot are directly ahead of each other. The selection of avoidance from the left or right direction is another major challenge taken in the proposed approach.

Görür and Erkmen [34] solved a related problem of social interaction between a person and the robot, where the robot was required to get attention of the person. The authors' predicted the person's trajectory and used the same for reshaping the trajectory separately in cases when the person was found confident and suspicious. Our approach models the avoidance behavior in corridor like scenarios where the intents are clear, without needing an estimation. Furthermore, in our case people are always confident and dominate the behavior and the robot reciprocates. We socially model the robot's navigation behavior under these assumptions.

Another set of approaches aim to integrate two or more than two collision avoidance strategies into a single model, such that the model retains the advantages of both the models. This fusion is done such that the advantages offered by the two techniques complement each other and do not work against each other. Ideally, a proactive and a reactive approach can be combined into a single model so that it can have the advantages of both the models in the resultant model. The extended forces model is used in conjunction with the reciprocal velocity obstacles algorithm where the former is used to deal with the navigation aspect of the problem while the latter deals with the human interactions and collision avoidance [35]. A modified social forces model considered the body pose and face orientation of the pedestrians in the surroundings to avoid collisions in a more human like manner [36]. The face orientation detection especially on a robot moving at near human speed is still an active research topic and the current algorithms are not very reliable.

2.3 Other Approaches

Learning based techniques aim to learn to emulate the human behavior from the human trajectory data and try to match the metrics such as socially acceptable distances [37]. The supervised learning approaches attempt to collect large amounts of data through teleoperation and then train a supervised training model using the collected data [38, 39]. Deep reinforcement learning is used to train an agent to learn to navigate safely and avoid collisions in an unknown environment [40, 41]. The reward function used to train a reinforcement learning model can be carefully designed to incorporate the social cues into the navigation mechanism by penalizing the non-desirable behaviors [42]. Inverse reinforcement learning is a technique that has been applied to learn a cost function from the human trajectory data and use it to train a reinforcement learning model [43–46]. Even autonomous navigation can be done for human conformability safety framework [47]. In addition, an offline path planning algorithm has been developed which is goal oriented and close to the bug algorithms [48]. Learning based methods are usually computationally intensive, require special hardware and undergo an expensive training process. Another limitation is the lack of interpretability of the solution provided by these techniques. The approaches may not generalize to all environments due to a bias towards the training data and training environments.

Some other algorithms from different schools of learning are described in the survey paper by Kruse [49]. In general, our aim is not to follow a deliberative path, but to use only a few sub-goals. This is because the humans will make the robot deviated by a very large amount. In our application further, most humans suddenly come ahead of the robot and move opposite to the robot. The little interaction time nullifies getting enough quality data for prediction.

Based on the literatures, it is evident that the investigation of social cues and their incorporation directly into the model-based approaches is currently absent. Even though methods have been used to make the approaches maintain social distances with humans, the larger cues of allowing overtaking on the right and avoiding on the left are absent from the literature. The learning-based approaches are not suited for the problem since the robots being socially and kinematically different from humans cannot simply imitate the exact human behavior. The learning of the human behavior is itself a hard task. Learning in the presence of humans and robots is a data and computationally intensive approach.

3 Social Potential Fields

The introduction of social cues for socially acceptable navigation can be formulated as extensions of existing decen-

tralized multiagent collision avoidance algorithms. In this section the social cues are introduced for social potential field approach.

3.1 Artificial Potential Field

The extended forces model assumes the presence of a virtual force field in the environment of the robot. The force field is made up of the different forces acting on the robot and consists of attractive forces from the goal, repulsive forces from the obstacles surrounding the robot and social forces from the humans. The robot is assumed to be equipped with a laser scanner whose readings are used by the algorithm to calculate the forces of repulsion. The magnitude of the forces applied on the robot are determined by the robot's location. Mathematically, the forces are modeled as follows. The force applied by the goal is proportional to the distance of the robot from the goal while the repulsive forces are inversely proportional to the square of the distance. The social forces accounting for social cues are dependent on the positions of the people in the environment within the vicinity of the robot.

Assume $q(q_x, q_y)$ is the position of the robot and $g(g_x, g_y)$ is the position of the goal. The attraction force is proportional to the distance to goal $d(q, g)$ and is directed towards the goal denoted as a unit vector in the direction $g - q$, or $\hat{u}(g - q)$. The distance is given a threshold of $d_{\max}$ to avoid excessively large values. The attraction is thus given by Eq. (1).

$$F_{attr} = \begin{cases} k_{att} d(q, g)\hat{u}(g - q) & if\ d(q, g) \leq d_{max} \\ k_{att} d_{max}\hat{u}(g - q) & if\ d(q, g) > d_{max} \end{cases} \quad (1)$$

k_{att} is the attraction proportionality constant. Consider θ_g be the angle to goal, shown by Eq. (2).

$$\theta_g = atan2\big(g_y - q_y, g_x - q_x\big) \quad (2)$$

This gives $\hat{u}(g - q)$ for a planar scenario as $[\cos(\theta_g)\ \sin(\theta_g)]^T$. The attraction force acting on the robot is thus given by Eq. (3). The notations are shown in Fig. 1.

$$F_{attr} = \begin{cases} k_{att} d(q, g) \begin{bmatrix} \cos(\theta_g) \\ \sin(\theta_g) \end{bmatrix} & if\ d(q, g) \leq d_{max} \\ k_{att} d_{max} \begin{bmatrix} \cos(\theta_g) \\ \sin(\theta_g) \end{bmatrix} & if\ d(q, g) > d_{max} \end{cases} \quad (3)$$

The repulsion force is applied by every obstacle o. The repulsive force is inversely proportional to the square of the distance $d(q, o)$ and is directed away from the goal, given by the unit vector $-\hat{u}(o - q)$. The repulsive force is given by Eq. (4).

$$F_{rep}(o) = -k_{rep} \frac{1}{d^2(q, o)} \hat{u}(o - q) \quad (4)$$

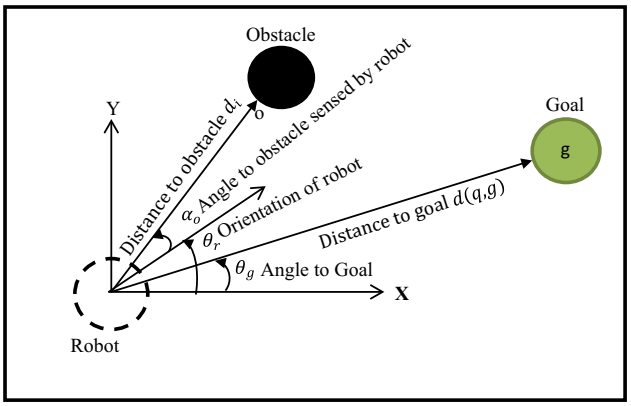


Fig. 1 Notations used for the calculation of attractive and repulsive forces. The attraction force is proportional to distance to goal (d_g) with an upper threshold, in the direction of goal (θ_g). The repulsive force is inversely proportional to the distance to obstacle (d_o), opposite to the direction to obstacle (α_o) with respect to heading direction of robot (θ_r)

Here k_{rep} is the repulsive force proportionality constant. It is assumed that the robot has a laser scanner and the ith laser hits at an angle of α_i from the heading direction of the robot (θ_r) to indicate a point obstacle at a distance d_i. The direction of the force is thus $[\cos(\alpha_i)\ \sin(\alpha_i)]^T$ in the robot's coordinate frame. If $T(R,W)$ is the transformation matrix between the robot and the world, the direction of force is given by $T(R,W)[\cos(\alpha_i)\ \sin(\alpha_i)]^T$. For this planar case, transform is a rotation along Z axis or $R_Z(\theta_r)$, and the force is given by Eq. (5)

$$\begin{aligned} F_{rep}(i) &= -k_{rep} \frac{1}{d_i^2} T(R, W) \begin{bmatrix} \cos(\alpha_i) \\ \sin(\alpha_i) \end{bmatrix} \\ &= -k_{rep} \frac{1}{d_i^2} R_Z(\theta_r) \begin{bmatrix} \cos(\alpha_i) \\ \sin(\alpha_i) \end{bmatrix} \\ &= -k_{rep} \frac{1}{d_i^2} \begin{bmatrix} \cos(\theta_r + \alpha_i) \\ \sin(\theta_r + \alpha_i) \end{bmatrix} \end{aligned} \quad (5)$$

The resultant repulsive force is the addition of repulsions of all obstacles detected by the laser scanner, given by Eq. (6)

$$F_{rep} = \sum_i F_{rep}(i) \quad (6)$$

3.2 Inducing Social Cues in Social Potential Field

Consider that two humans are facing each other with their goals in the opposite direction and want to cross each other to reach their goals. In a potential field approach, the attraction of the person A towards its goal and the repulsion force applied on person A by person B will cancel each other out as shown in Fig. 2. In reality, little errors at the end cause the symmetry to be broken and eventually the humans avoid each other rather than colliding.

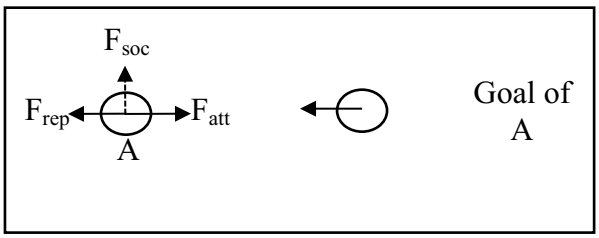


Fig. 2 Equilibrium condition without the social force when two humans walk opposite to each other

However, in such a situation, the human *A* practically sees the other human at the front and starts to change the moving direction much earlier in anticipation to avoid collision. Consider a narrow corridor with two humans walking towards each other. On looking at each other, the humans would in-anticipation drift to the left much earlier, rather than the simulated behavior with the potential field where they avoid each other right at the end. To get the same effect in the robots, an orthogonal force is proposed as shown in Fig. 2.

The social force in the extended force model picks a direction to move well in advance and the human is expected to reciprocate the behavior by picking the opposite direction to move, thus cooperating to avoid collisions. Overtaking is another behavior, while common to humans is uncommon to mobile robots because they usually move at much lesser than the human walking speed. This results in a consequent behavior of being overtaken modeled in the robot. If a human is detected behind the robot, then a similar social force comes into play allowing the human to overtake. The robot drifts at one side to allow easy overtaking by the human from the other side.

Humans generally tend to follow a convention when it comes to cooperative collision avoidance strategies. Passing on the left is followed according to the British convention and the same rule can be reversed according to the American convention. When a human is detected walking towards the robot, the social force model applies a force towards the left side, which is inversely proportional to the distance of the human from the robot and the robot picks the left side to pass the human. Similarly, in presence of a human walking towards the robot from behind, a similar force is applied to the left side allowing the human to overtake from the right-hand side. The problem of oscillatory behavior is therefore removed with the introduction of the social cues and safer distances are maintained from the pedestrians to not invade their personal space.

Consider a person p at a distance of $d(q, p)$ from the robot at position q. The social force by the person p is inversely proportional to the square of the distance between the robot and person $d(q, p)$. The force is applied orthogonal to the direction to person. Unlike some literatures, the proposed approach does not treat the persons as they were moving obstacles. The force due to people as regular obstacles is added opposite to the direction of the obstacle, and additionally, for people a social force is added orthogonal to the direction to the person. This results in the approach forcing the person on the left or right, depending upon the socially selected side, at each and every iteration. This force results in enough clearance and safety for the robot.

The direction to person is given by $\hat{u}(p-q)$, and a orthogonal direction (leftward) has an additional rotation of $\pi/2$ in Z axis, or $R_Z\left(-\frac{\pi}{2}\right)\hat{u}(p-q)$. The resultant force is thus given by Eq. (7).

$$F^L_{soc}(p) = -k_{soc}\frac{1}{d^2(q,p)}R_Z\left(-\frac{\pi}{2}\right)\hat{u}(p-q) \tag{7}$$

Similar to obstacles, modelling the unit vector $\hat{u}(p-q)$ by the relative angle between person and robot (θ^R_p) and further transforming in the word coordinate frame by multiplication with $T(R, W)$, gives an expression similar to the obstacles shown by Eq. (8).

$$\begin{aligned} F^L_{soc}(p) &= -k_{soc}\frac{1}{d^2(q,p)}R_Z\left(-\frac{\pi}{2}\right)T(R,W)\begin{bmatrix}\cos\left(\theta^R_p\right)\\ \sin\left(\theta^R_p\right)\end{bmatrix} \\ &= -k_{soc}\frac{1}{d^2(q,p)}R_Z\left(-\frac{\pi}{2}\right)R_Z(\theta_r)\begin{bmatrix}\cos\left(\theta^R_p\right)\\ \sin\left(\theta^R_p\right)\end{bmatrix} \\ &= -k_{soc}\frac{1}{d^2(q,p)}\begin{bmatrix}\cos\left(\theta_r+\theta^R_p-\frac{\pi}{2}\right)\\ \sin\left(\theta_r+\theta^R_p-\frac{\pi}{2}\right)\end{bmatrix} \end{aligned} \tag{8}$$

To avoid the person in a rightward direction, the force vector is rotated by $-\pi/2$ in Z axis instead of $\pi/2$ in Z axis, given by Eq. (9)

$$F^R_{soc}(p) = -k_{soc}\frac{1}{d^2(q,p)}\begin{bmatrix}\cos\left(\theta_r+\theta^R_p+\frac{\pi}{2}\right)\\ \sin\left(\theta_r+\theta^R_p+\frac{\pi}{2}\right)\end{bmatrix} \tag{9}$$

The social force is typically applied in the leftward direction. However, the avoidance should be on the right-hand side instead if the person is already sufficiently on the left. The robot cannot go far on the left to avoid the person already on the left-hand side.

Let θ^R_p denote the orientation of the person as observed from the robot. If θ^R_p is 0, it means that the robot is facing the person head-on and hence avoidance should be on the left. If θ^R_p is highly positive, it means that the person is on the left and the robot should avoid the person from the right. If is nearly 0 or highly negative, the person should be avoided from the left. A limiting value of θ_{left} is given to choose between the

2 strategies. The resultant force is given by Eq. (10). The notations are shown in Fig. 3.

$$F_{soc}(p) = \begin{cases} F^L_{soc}(p) & if\ \theta^R_p \le \theta_{left} \\ F^R_{soc}(p) & if\ \theta^R_p > \theta_{left} \end{cases} \tag{10}$$

The resultant force is the vector addition of all social forces due to all people, given by Eq. (11)

$$F_{soc} = \sum_p F_{soc}(p) \tag{11}$$

The social force model consists of three types of forces: (1) repulsive force from the laser scan which gives the distance readings of the obstacles; (2) attractive force towards the goal; and (3) social forces enforcing to the conventions to be followed in presence of persons. The resultant force is the summation of the attractive, repulsive and social forces given by Eq. (12).

$$F_{res} = F_{attr} + F_{rep} + F_{soc} \tag{12}$$

3.3 Overall Approach

The robot aims to move in the direction of the resultant force vector or $atan2(F_y, F_x)$. The angular error is the difference between the robot's orientation and intended direction of motion. Using a proportionate controller, the angular speed is given by Eq. (13)

$$\omega_{\text{pref}} = K_\omega angleDifference\left(atan2\left(F_y, F_x\right), \theta_r\right) \tag{13}$$

Algorithm 1 - Social Potential Fields with Social Cues

Input: *laser scan* $(D = \{< d_i, \alpha_i >\})$, *front people* $(P_F = \{p_f\})$, *rear people* $(P_R = \{p_r\})$, *robot pose* $q(x_r, y_r, \theta_r)$, $goal(g)$

Output: *linear speed* (v) *and angular speed* (ω)

Method:

1. $F \leftarrow [0\ 0]^T$
2. *For* $< d_i, \alpha_i > \epsilon\ D$ *do*
   a. $F \leftarrow F - k_{rep}\frac{1}{d_i^2}\begin{bmatrix}\cos(\theta_r + \alpha_i)\\ \sin(\theta_r + \alpha_i)\end{bmatrix}$
3. *If* $(d(q,g) \le d_{max})$ *then*
   a. $F \leftarrow F + k_{att} d(q,g)\begin{bmatrix}\cos(\theta_g)\\ \sin(\theta_g)\end{bmatrix}$
4. *else*
   a. $F \leftarrow F + k_{att} d_{max}\begin{bmatrix}\cos(\theta_g)\\ \sin(\theta_g)\end{bmatrix}$
5. *For* $p \epsilon\ P_{rear} \cup P_{front}$ *do*
   a. *if* $\theta^R_p \le \theta_{left}$ *then*
      i. $F \leftarrow F - k_{soc}\frac{1}{d^2(q,p)}\begin{bmatrix}\cos\left(\theta_r + \theta^R_p - \frac{\pi}{2}\right)\\ \sin\left(\theta_r + \theta^R_p - \frac{\pi}{2}\right)\end{bmatrix}$
   b. else
      i. $F \leftarrow F - k_{soc}\frac{1}{d^2(q,p)}\begin{bmatrix}\cos\left(\theta_r + \theta^R_p + \frac{\pi}{2}\right)\\ \sin\left(\theta_r + \theta^R_p + \frac{\pi}{2}\right)\end{bmatrix}$
6. $\omega_{\text{pref}} = K_v angleDifference\left(atan2\left(F_y, F_x\right), \theta_r\right)$
7. *if* $\omega_{pref} \le 0$ *then*
   a. $\omega \leftarrow \max\left(\omega_{pref}, -\omega_{max}\right)$
8. else
   a. $\omega \leftarrow \min\left(\omega_{pref}, \omega_{max}\right)$
9. $v = K_\omega \min\left(d_{front}(q), d(q,g)\right)$
10. *return* (v, ω)

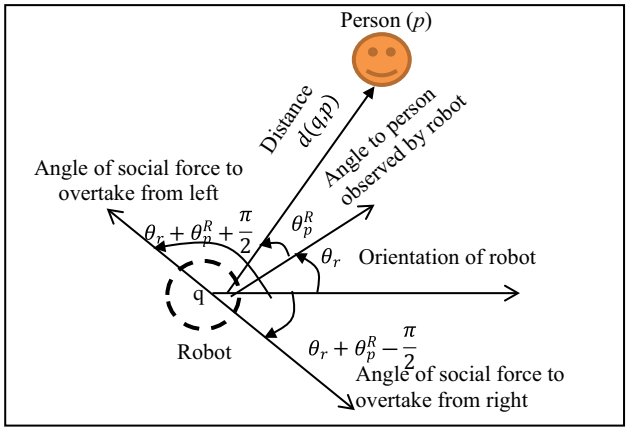


Fig. 3 Notations used for the social forces. The social force due to person is inversely proportional to square of distance, and is directed in either of the angles shown, orthogonal to the vector to person

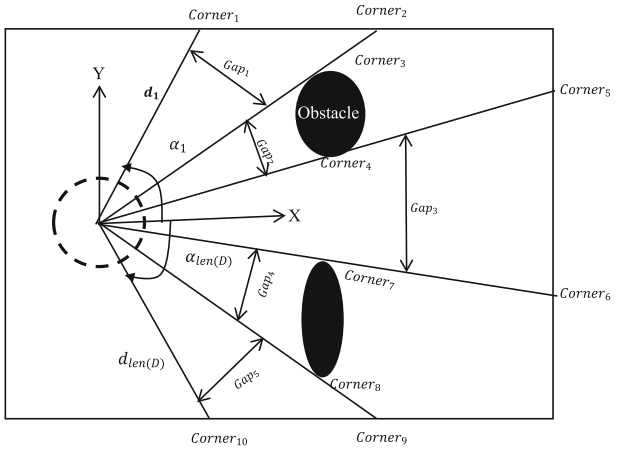


Fig. 4 Gaps for a general environment. Every angle at which the laser value drastically changes is called as the corner, including 1st and last laser reading at α_1 and $\alpha_{len(D)}$. A gap is the space between every corner

Here ω_{pref} is the preferred angular velocity. K_ω is the controller gain parameter. Applying the limits of $[\omega_{max}, -\omega_{max}]$ in the acceptable range for angular speed gives the speed as Eq. (14).

$$\omega = \begin{cases} \max(\omega_{pref}, -\omega_{max}) & if\ \omega_{pref} \leq 0 \\ \min(\omega_{pref}, \omega_{max}) & if\ \omega_{pref} > 0 \end{cases} \tag{14}$$

The linear speed is kept so as to avoid head-on collision and not to surpass the goal. Using a proportionate controller, the speed is given by Eq. (15).

$$v = K_\omega \min\left(d_{front}(q), d(q,g)\right) \tag{15}$$

K_v is the controller gain parameter.

The algorithm has a set of parameters which are easy to set. The parameters are tuned based on a socially acceptable distance which is obtained from a set of trials with the robot. The parameters are set on the basis that equilibrium is obtained and the forces cancel each other out at the socially acceptable distance which is set as approximately 0.5 meters. The working of the algorithm is shown by Algorithm 1.

4 Gap Based Strategy

Geometric approaches are a class of collision avoidance algorithms that are based on locating the obstacle free regions in the robot's surroundings and picking the best region to pass through. The Follow the Gap method picks the largest gap in the robot's surroundings and moves towards the gap. This strategy however does not account for the direction of the gap with respect to the direction of goal. It can sometimes lead to long and inefficient trajectories if the largest gap found is facing away from the direction of the goal. Even when the direction to the goal is accounted for in the gap selection strategy, the method is prone to oscillatory behavior even for very small changes in direction. The idea is to break down the gap selection strategy into a two-way qualification and selection process where the qualification strategy accounts for the size of the robot and its dynamics while the selection strategy accounts for the social cues. Unlike some literatures, the people are not treated as static obstacles. The social cues govern the selection of the gap. The robot may not select the best gap if it is not socially compliant.

4.1 Computing Gaps

The first aspect is to calculate the corners in the map using a laser scan. For a sensor that reports unbound distances, a corner is defined at a point where the distance abruptly changes on a small change in the sweeping angle. Let d_i and be the current and d_{i-1} be the previous laser reading (similarly next). If the absolute difference, $abs(d_i - d_{i-1})$ with the previous reading is high (similarly next), then ith laser has hit a corner at distance d_i and angle α_i, which represents a vector $d_i[\cos(\alpha_i)\ \sin(\alpha_i)]^T$ with respect to the robot. Corners are an ordered sequence of all such points, given by Eq. (16).

$$corners = \left[d_i \begin{bmatrix} \cos\alpha_i \\ \sin\alpha_i \end{bmatrix} : abs(d_i - d_{i-1}) \geq \epsilon \vee abs(d_i - d_{i+1}) \geq \epsilon \vee i = 1 \vee i = len(D) \right] \tag{16}$$

Here [] is used to denote an ordered list. $len(D)$ is the size of the laser scan. There are two exceptional cases, which is the first laser reading ($i = 1$) and last laser reading ($i = \text{len}(D)$) in a laser scan with bounded field of view. They are both taken as corner points. ε is a corner acceptance threshold. Every abrupt change registers two corners at two consecutive laser readings. The corners are shown in Fig. 4.

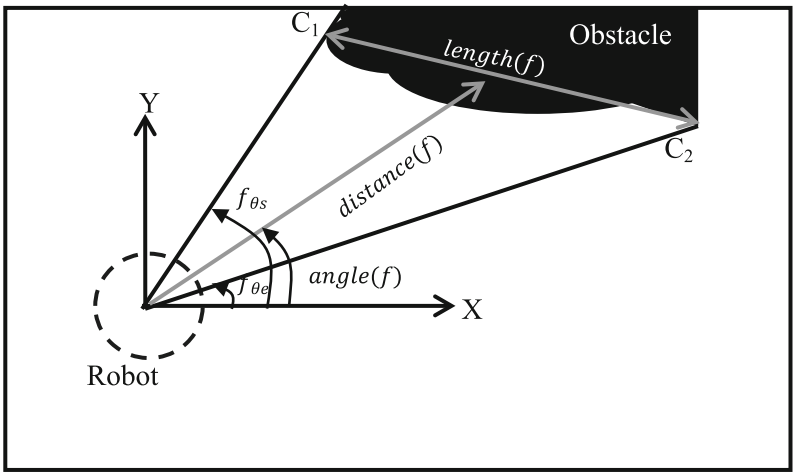


Fig. 5 Properties of gaps. A gap (f), detected by a laser scan between angles $f_{\theta s}$ and $f_{\theta e}$ is attributed by a central angle ($angle(f)$), distance to gap ($distance(f)$), and length of gap ($length(f)$)

The gaps are the open spaces between two consecutive corners. A gap (f) is characterized by the two corners (f_{C1} and f_{C2}) with gap as the space between the two corners. The gap array is a collection of all consecutive corners. Since one laser scan gives two corners, the gaps are taken between all odd indexed corner with the next consecutive corner. The set of gaps are given by Eq. (17).

$$gaps = \{\langle f_{C1}, f_{C2}\rangle\} = \{\langle corners(i), corners(i+1)\rangle : i = 1, 3, 5 \ldots len(corners) - 1\} \quad (17)$$

The gaps are shown in Fig. 4. A gap (f) is associated with 3 properties. The properties are shown in Fig. 5. $length(f)$ denotes the space available to go through the gap. The length is the distance between the corners that make the gap, given by Eq. (18).

$$length(f) = d(f_{C1}, f_{C2}), f\langle f_{C1}, f_{C2}\rangle \in gaps \quad (18)$$

$distance(f)$ denotes the average distance of the gap from the robot. Every laser angle between the two corners gives a distance, which averaged gives the distance between the robot and gap, given by Eqs. (19–21).

$$distance(f) = \frac{\sum_{i:f_{\theta s} \le \alpha_i \le f_{\theta e}} d_i}{\sum_{i:f_{\theta s} \le \alpha_i \le f_{\theta e}} 1}, f\langle f_{C1}, f_{C2}\rangle \in gaps \quad (19)$$

$$f_{\theta s} = atan2\left(f_{c1,y}, f_{c1,x}\right) \quad (20)$$

$$f_{\theta e} = atan2\left(f_{c2,y}, f_{c2,x}\right) \quad (21)$$

Here $f_{\theta s}$ is the start of the gap angle calculated by the angle to the first corner, and $f_{\theta e}$ is the end of the gap angle calculated by the angle to the second corner. The gap strictly lies between the angles $f_{\theta s}$ and $f_{\theta e}$.

The gap mean angle is denoted by $angle(f)$ and is given by a weighted average of all angles, weighted by the distance. The formula is similar to the computation of the centre of area. The gap mean angle is given by Eq. (22).

$$angle(f) = \frac{\sum_{i:f_{\theta s} \le \alpha_i \le f_{\theta e}} d_i \alpha_i}{\sum_{i:f_{\theta s} \le \alpha_i \le f_{\theta e}} d_i}, \langle f_{C1}, f_{C2}\rangle \in gaps \quad (22)$$

4.2 Selecting Best Gap

The gaps are filtered by a qualification process to pick those set of gaps which are acceptable ($gaps_Q$). The qualification occurs based on the size of the gap and the size should be at least twice the radius of the robot for it to be an acceptable gap. This allows the robot to pass through. Further, gaps too close to the robot are undesirable and act as obstacles. The qualification criterion of gaps is hence given by Eq. (23).

$$gaps_Q = \{f \in gaps : length(f) \ge 2R \wedge distance(f) \ge \eta\} \quad (23)$$

Here R is the radius of the robot and η is the gap acceptance threshold. The equation considers all gaps in the gap set, and makes a new set $gapsQ$ which have a length of at least $2R$ and are at a distance of at least η.

After the qualification process, the final gap selected is on the basis of being closer to the direction of goal. The gap is at $angle(f)$ with respect to the robot, which is oriented at θ_r with respect to the world. The direction of the gap with respect to the world is thus at $angle(f) + \theta_r$, while the direction to goal is θ_g. The angular error between the gap and goal is thus. $angleDifference(angle(f) + \theta_r - \theta_g)$. The gap with the least angular error is used. The best gap (f_2) is thus given by Eq. (24).

$$f_1 = \arg\min_{f \in gaps_Q} angleDifference\left(angle(f) + \theta_r - \theta_g\right) \quad (24)$$

Here θ_r is the orientation of the robot and is added to convert the gaps in the robot frame of reference to the world frame of reference. θ_g is the angle to goal.

The selection strategy is improved with the introduction of the social cues in an intuitive manner. If there are two equally likely gaps on both sides of the goal, then the gap picked will be the one on the left-hand side. However, if the nearest person is already sufficiently on the left, similar to the potential field, the robot does not try to avoid it from the left side. In this way, the oscillatory behavior is avoided with the help of the social cues and the extra parameter required for calculating the heading angle in the follow the gap method is removed.

Let f_2 be the 2nd best gap given by Eq. (25)

$$f_2 = \underset{f \in gaps_Q, f \neq f_1}{\arg\min} \; angleDifference\big(angle(f) + \theta_r - \theta_g\big) \tag{25}$$

The selected gap is hence given by Eq. (26).

$$f_{soc} = \begin{cases} left(f_1, f_2) & if\ abs(angle(f_1) - angle(f_2)) \leq \nu \wedge \theta_p^R \leq \theta_{left} \\ right(f_1, f_2) & if\ abs(angle(f_1) - angle(f_2)) \leq \nu \wedge \theta_p^R > \theta_{left} \\ f_1 & otherwise \end{cases} \tag{26}$$

Here ν is the threshold within which a social selection strategy is invoked. If the best and 2nd best gaps are both competent, then the angular difference between their angles $abs(angle(f_1) - angle(f_2))$ is small. In such a case the social cues are followed, otherwise only the best gap is used. p is the nearest person and θ_p^R is the angle to the nearest person in the frame of reference of the robot. If the person is sufficiently right, then the avoidance is by right, otherwise, a left avoidance is preferred. The function *left*() selects the leftward of the two input angles as a socialistic behavior and *right*() selects the rightward of the two input angles.

Algorithm 2 - Gap Qualification Strategy

$Input$: $laserscan\ D = [d_i, \alpha_i]$

$Output$: $qualified\ gaps$: $gaps_Q$

$Method$:

1. $corners \leftarrow [0]$
2. $For\ i\ \epsilon\ \{1,2,\ldots,len(D)\}\ do$
   a. $If\ (abs(d_i - d_{i-1}) \leq \epsilon)\ then$
      i. $corners.append(i-1)$
      ii. $corners.append(i)$
3. $corners.append\big(len(D)\big)$
4. $gaps_Q = []$
5. $For\ each\ gap\ f\ characterized\ by\ adjacent\ corner\ points\ (i,j) do$
   a. $length(f) = d(f_{C1}, f_{C2})$
   b. $s_{distance} \leftarrow 0$
   c. $s_{angle} \leftarrow 0$
   d. $For\ k \in \{corners[i], corners[j]\}\ do$
      i. $s_{distance} \leftarrow s_{distance} + d_k$
      ii. $s_{angle} \leftarrow s_{angle} + \alpha_k d_k$
   e. $angle(f) = \frac{s_{angle}}{s_{distance}}$
   f. $distance(f) = \frac{s_{distance}}{abs(corners[i] - corners[j]+1)}$
   g. $If\ (length(f) \geq 2R\ \wedge distance(f) \geq \eta) then$
      i. $gaps_Q.append(f)$
6. $return\ (gaps_Q)$

4.3 Overall Approach

Given a gap f_{soc}, the aim is to fuse motion to goal behavior asking the robot to go directly towards the goal and going towards the chosen gap. Nearer to the obstacles, the seeking gap behavior dominates and away from the obstacles, the goal seeking dominates. The preferred angle is given by Eq. (27)

$$\theta_{gap} = \frac{\frac{w}{\min(d_i)} angle(f_{soc}) + \theta_g}{\frac{w}{\min(d_i)} + 1} \tag{27}$$

Here $min(d_i)$ is the distance to the closest obstacle. w is an algorithm parameter such that $\frac{w}{\min(d_i)}$ is the relative weight of the gap seeking behavior.

The algorithm for the gap selection is shown as Algorithm 2. The gap qualification strategy first extracts the corner points from the laser scan by taking the derivative of the adjacent points. The gaps are then obtained from the corner points based on a qualification criterion. The gap size must be larger than two times the size of the robot for it to qualify as an acceptable gap and the gap must be far enough. A deviation from the mathematical formulation is that the laser indices are used to denote the corners.

Algorithm 3 - Gap Selection Strategy

Input: *qualified gaps*: $gaps_Q$, *front people* $(P_F = \{p_f\})$, *rear people* $(P_R = \{p_r\})$, *robot pose* $q(x_r, y_r, \theta_r)$, $goal(g)$

Output: *selected angle of motion* θ_{des}

Method:

1. $f_1 = \arg\min_{f \in gaps_Q} angleDifference\big(angle(f) + \theta_r - \theta_g\big)$
2. $f_2 = \arg\min_{f \in gaps_Q, f \neq f_1} angleDifference\big(angle(f) + \theta_r - \theta_g\big)$
3. $p \leftarrow \arg\min_{\mathrm{p} \in \mathrm{P}_{\mathrm{front}} \cup P_{rear}} d(q, p)$
4. $\theta_p^R \leftarrow orientation\ of\ person\ wrt\ the\ robot$
5. $If\big(len\big(P_{front} \cup P_{rear}\big) \geq 1\big) then$
   a. $If abs\big(angle(f_1) - angle(f_2)\big) \leq \nu\ then$
      i. $If\ \big(\theta_p^R \leq \theta_{left}\big)\ then$
         (A). $f \leftarrow left(f_1, f_2)$
      ii. $Else$
         (B). $f \leftarrow right(f_1, f_2)$
   b. $Else$
      i. $f_{soc} \leftarrow f_1$
6. $Else$
   a. $f_{soc} \leftarrow f_1$
7. $If\big(d(g, q) \leq distance(f_{soc})\big)\ then$
   a. $\theta_{gap} \leftarrow \theta_g$
8. $Else$
   a. $\theta_{gap} \leftarrow \frac{\frac{w}{\min(d_i)} angle(f_{soc}) + \theta_g}{\frac{w}{\min(d_i)} + 1}$
9. $return\ \theta_{des}$

The gap selection is shown by Algorithm 3. The gap selection strategy processes the available gaps to select the ideal gap. The gap is chosen based on its direction with respect to the direction to the goal and the gap whose direction is closest to that of the goal is picked. The social cues are accounted for by giving a preference to the left-hand side when two gaps which are equally close to the goal are available in the presence of a human. A threshold is applied like the potential field. If the gap is after the goal, it means a straight-line motion to goal is possible and the robot selects the same.

5 Hybrid Force Model

The extended forces model accounts for the cooperation between the robot and the person through social cues; but the approach is reactive, meaning that the action taken to avoid the obstacle is after the obstacle is close enough to the robot. Reactive approaches tend to suffer from the problem of being stuck in a local minimum because they attempt to avoid collision only once the obstacle is sufficiently close enough to the robot. The extended gap method, while still a local planning algorithm, is more predictive compared to the force model. The robot attempts to move through obstacle free regions thus being more proactive by moving in obstacle free regions. The robot takes predictive actions to avoid the collisions. The quality of the trajectory generated however is dependent on the quality of the gaps detected which can sometimes be a tricky task. The lack of a reactive strategy in this method can sometimes lead to the robot generating a trajectory which moves too close to the human therefore encroaching their personal space and making it uncomfortable.

5.1 Fusion Design

The idea is to combine the reactive nature of the social force model with the proactive nature of the extended follow the gap method, such that the characteristics of both the methods are retained by the hybrid method. The qualification and selection strategy are integrated into the force model whose actions have both predictive and reactive properties. The predictive property is derived from the gap method where the robot tries to find the best obstacle free region and the reactive property is derived from the force model where the repulsion is very high when the obstacle is close and therefore preventing collisions.

The integration can occur in two ways. As a naïve way to integrate, the resultant angle of the force model can be combined with the gap angle using a single parameter to calculate the final heading angle which acts as the trade-off between the two methods. Consider F as the resultant total force, urging the robot to move in the direction $\theta_{\mathrm{APF}} = atan2(F_{\mathrm{y}}, F_{\mathrm{x}})$. Let the desired angle by the follow the gap method be $\theta_{\mathrm{gap}} = angle(f_{\mathrm{sel}})$. The desired resultant angle may thus be given by $(w_{\mathrm{APF}}\ \theta_{\mathrm{APF}} + w_{\mathrm{gap}}\ \theta_{\mathrm{gap}})$, where w_{APF} and w_{gap} are the weights associated with the strategies. There are two problems with this approach. The first is that potential fields and follow the gap can have contradictory ways to avoid the obstacle, whose fusion can result in a collision. Say an obstacle somewhat ahead is seen. The first method suggests a leftward avoidance, while the second suggests a rightward avoidance, whose fusion results in a collision. Secondly, the relative weights are very hard to set.

The technique used is to integrate the selected gap directly into the force model as an attractive force, where the gap is modeled as a sub-goal to the main goal. No extra parameters are required in this method as the force from the gap/sub-goal replaces the attractive force from the main goal. The force from the main goal only comes into play when the distance from the destination to the robot is lesser than that of the selected gap from the robot. The gap selected by the selection strategy acts as a sub-goal for the robot and the attractive force is applied from the gap if it is closer to the robot compared to the goal. The attractive force is thus given by Eq. (28).

$$F_{attr} = \begin{cases} k_{att} d(q,g) \begin{bmatrix} \cos(\theta_{gap}) \\ \sin(\theta_{gap}) \end{bmatrix} & if\ d(q,g) \leq d_{max} \\ k_{att} d_{max} \begin{bmatrix} \cos(\theta_{gap}) \\ \sin(\theta_{gap}) \end{bmatrix} & if\ d(q,g) > d_{max} \end{cases} \tag{28}$$

Here θ_{gap} is the angle to gap as calculated by algorithm 3.

The social forces act in the same manner as previously in the force model. The social cues followed in the gap selection strategy works with the social forces to compound each other and thus result in safer trajectories. In this way, the gap strategy is integrated into the social force model in a very intuitive

manner without any additional parameters and combines the proactive and reactive natures of both the methods.

Algorithm 4 - Hybrid Potential field algorithm

$Input$: $laserscan\ (D = [d_i, \alpha_i])$, $selected\ gap\ angle\ (\theta_{gap})$, $qualified\ gaps: gaps_Q$, $front\ people\ (P_F = \{p_f\})$, $rear\ people\ (P_R = \{p_r\})$, $robot\ pose\ q(x_r, y_r, \theta_r)$, $goal(g)$

$Output$: $angle\ of\ the\ motion$

$Method$:

1. $F \leftarrow [0\ 0]^T$
2. $For < d_i, \alpha_i > \epsilon\ D\ do$
   a. $F \leftarrow F - k_{rep} \frac{1}{d_i^2} \begin{bmatrix} \cos(\theta_r + \alpha_i) \\ \sin(\theta_r + \alpha_i) \end{bmatrix}$
3. $If (d(q, g) \leq d_{max})\ then$
   a. $F \leftarrow F + k_{att} d(q, g) \begin{bmatrix} \cos(\theta_{gap}) \\ \sin(\theta_{gap}) \end{bmatrix}$
4. $else$
   a. $F \leftarrow F + k_{att} d_{max} \begin{bmatrix} \cos(\theta_{gap}) \\ \sin(\theta_{gap}) \end{bmatrix}$
5. $For\ p \epsilon\ P_{rear} \cup\ P_{front}\ do$
   a. $if\ \theta_p^R \leq \theta_{left}\ then$
      i. $F \leftarrow F - k_{soc} \frac{1}{d^2(q,p)} \begin{bmatrix} \cos\left(\theta_r + \theta_p^R - \frac{\pi}{2}\right) \\ \sin\left(\theta_r + \theta_p^R - \frac{\pi}{2}\right) \end{bmatrix}$
   b. else
      i. $F \leftarrow F - k_{soc} \frac{1}{d^2(q,p)} \begin{bmatrix} \cos\left(\theta_r + \theta_p^R + \frac{\pi}{2}\right) \\ \sin\left(\theta_r + \theta_p^R + \frac{\pi}{2}\right) \end{bmatrix}$
6. $\omega_{\text{pref}} = K_v angleDifference\left(atan2\left(F_y, F_x\right), \theta_r\right)$
7. $if\ \omega_{pref} \leq 0\ then$
   a. $\omega \leftarrow \max\left(\omega_{pref}, -\omega_{max}\right)$
8. else
   a. $\omega \leftarrow \min\left(\omega_{pref}, \omega_{max}\right)$
9. $v = K_\omega \min\left(d_{front}(q), d(q, g)\right)$
10. $return\ (v, \omega)$

5.2 Overall Approach

The working of the algorithm is given in Algorithm 4. The social forces present in the extended force model remain unchanged and they come into play when people are detected moving towards the robot with the help of the cameras. The passing on the left convention is followed by default and can be easily changed by reversing the directions of the forces applied in case of a different convention. The selection strategy accounts for the social cues as well by factoring them when choosing the best gap/sub-goal for the robot to move into. As the conventions followed by both the strategies are the same, their effects compound each other. For instance, when a person is walking towards the robot, the gap selection strategy enforces the convention and picks the gap on the left side of the person as its sub-goal. The social force arising from the person's presence is also directed to the left. The resultant of the forces ensures that the robot will pass on the left-hand side and maintain a safe distance from the person. In the potential field approach the robot moves forward until the pedestrian is at a sufficient distance, at which point it reacts to his or her presence. In the hybrid strategy, the robot starts moving towards a gap in a proactive manner and then makes a maneuver to maintain a safe distance from

the person. In this way, the robot moves in a more human-like manner as well as maintains a larger distance from the pedestrians thereby respecting their private space.

6 Implementation

The system is implemented on the Pioneer LX robot. The Pioneer LX is equipped with a laser scanner to detect the obstacles around. Additionally, two Logitech webcams were mounted on top of the robot, one in front and one at the rear, in order to detect the presence of humans. This section discusses the different modules that complete the overall system architecture. The overall architecture representing the approach is shown in Fig. 6.

6.1 Face Detection and Tracking

The robot is attached with a front and a rear camera which are used to perform face detection for the presence of humans. The facenet model [50], which is the state-of-the-art model for face detection and recognition, is used for this purpose. The face detection problem [51] is decomposed into multiple stages for robust recognition. As the first stage, a convolution neural network is used to generate several candidate person detections along with their bounding boxes. This is called as the P-Net. A second network refines the detections and is thus called as the R-Net. The output is finally processed by an output network called as the O-Net.

A limitation was observed that the Logitech web camera does not perform very well when it is in motion, which in turns contributes to a poor performance of the face detection model. Hence, a correlation tracker is used to track the detected face for a certain number of frames, before performing detection once again.

6.2 Face Distance and Angle Computation

Once the bounding box of the face is obtained, the area of the bounding box is used to calculate the distance and orientation of the person from the robot. The area of the bounding box is observed to be inversely proportional to the distance of the person from the robot.

A sample data was collected by placing subjects at different distances from the robot and then measuring the area of the bounding box detected. The angle of the person was observed as well. The camera had a 60-degree field of view and pixel location of the bounding box was used to calculate the angle. A linear regression model was trained on the collected data to predict the distance of a person, given the size of the bounding box of his/her face and its orientation. The model gave acceptable results upon further testing and was hence used to detect the positions of the humans.

6.3 Sequencing

The problem of reaching a goal directly may not be possible in a reactive methodology because of the presence of complex obstacles on the way. The reactive algorithms can often get stuck in complex obstacle scenarios. Therefore, the problem of motion to goal is decomposed by breaking a deliberative path to goal into several sub-goals. The reactive control algorithms are only provided with the sub-goals. As soon as the robot reaches a sub-goal, within a tolerance level, the sequencer provides the next sub-goal to the robot. In this way, on progressing forward, the robot keeps getting the next sub-goals, until it reaches near enough to the goal, when it is given the goal.

For testing purposes, the robot had to be operated for long runs within a limited space. Therefore, a sequencer was further made that updates the goal of the robot, as soon as the robot accomplishes a goal. In this way, the robot was programmed to cover a number of goals in a sequence, while socially avoiding the people on the way.

The implementation of the sequencer was done using a Behavioral Finite State Machine (BFSM). The machine could take two states, waiting for the robot to reach the goal and goal reached. The transition from the waiting state to the goal reached was based on a feedback from the controller. In the waiting state, the BFSM did not do any further computation and just waited for the controller to send a feedback. In the goal reached state, the BFSM computed the next goal to be given to the robot, which was given to the controller and the BFSM immediately moves to the waiting state.

The overall system was implemented using the Robot Operating System (ROS) [52]. The different modules executed as different nodes of ROS, which communicated via ROS messages. A representative ROS graph is given by Fig. 7.

7 Results

The different approaches were tested across multiple scenarios in both the simulations and on a real robot and the distance maintained from the humans was used as a metric to compare the performances of the approaches.

7.1 Evaluation Metric

A penalty metric is designed factoring in the socially acceptable distance. Penalties are given to the robot through the course of the trial when it violates the distance constraint. The metric is given by Eq. (29).

$$\text{penalty(t)} = \begin{cases} \frac{0.1}{d(q,p)} & if\ d(q,p) \leq d_{soc} \\ 0 & otherwise \end{cases} \tag{29}$$

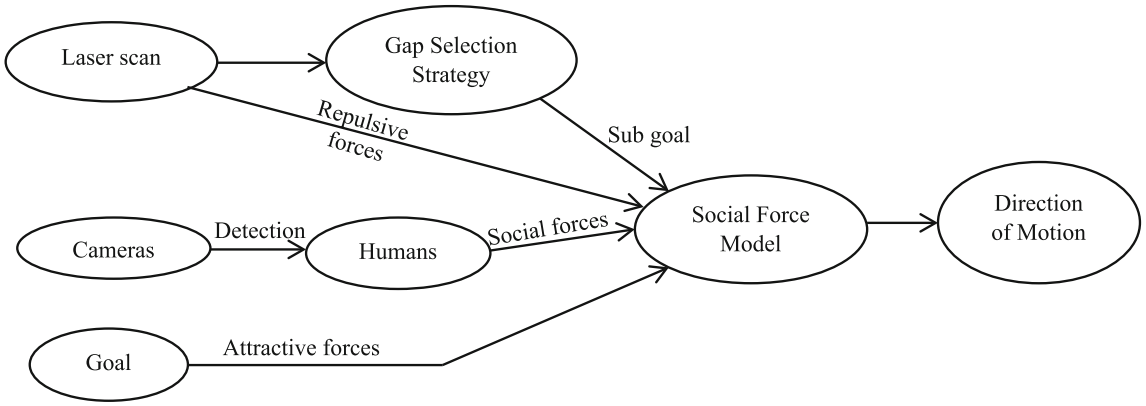


Fig. 6 Overall architecture of the hybrid model. The laser scan gives information about obstacles, used to calculate repulsive forces. The humans (detected by camera) are used to calculate social forces. The attraction forces are calculated by the goal. The local map produced by laser scan gives the gap, which produces a sub-goal for attraction. The forces determine direction of motion

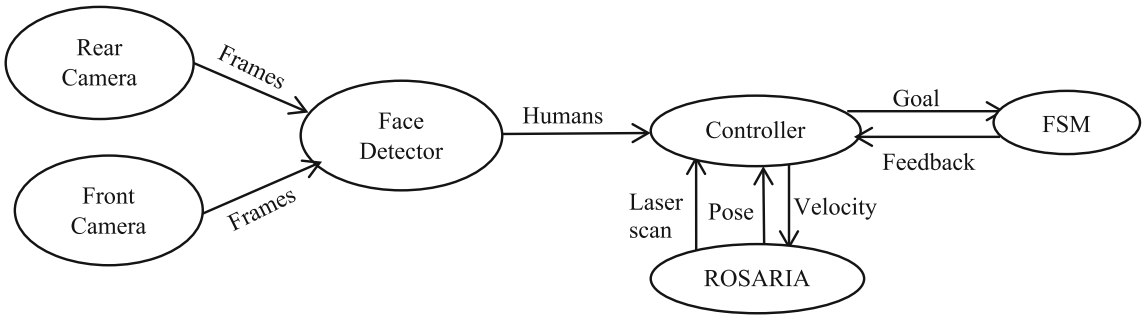


Fig. 7 ROS Graph for the problem. The vertices represent software modules. The edges represent the information interchanged. The ROSARIA is the manufacturer supplied software suite for interacting with the robot

Here d_{soc} is the maximum social distance as observed by humans, which was taken as 0.5 m. The metric of 0.5 m was derived as follows. Experiments were done by observing humans in natural environments. A large room was used for the study, so that the environmental bias is extremely small. People were made to avoid each other. The people were tracked using a Quanergy 3D lidar mounted on a tripod. The 3D image at every time step was processed to remove background, remove outliers, perform orthographic projection, and cluster to separate the individuals. The detected individuals were then subjected to filtering by using a Kalman filter. The separation of people at different times, specifically during avoiding each other were noted. The experiments revealed an avoidance clearance of 0.2401 m $\pm$ 0.0382 m. In our experiments, the distance between the centres of the robot and person is used as the clearance. Thus, the desired clearance is (approximately) 0.2401 m (clearance) + 0.25 m (radius of the robot) = 0.4901 m. Note that the separation is measured by a lidar which touches the obstacle (person) and therefore the radius of the person is not excluded. An additional factor of 0.01 m is left since the human may not be as comfortable with the robot as another human.

The four approaches used for testing are given as follows. The APF method is the basic social force model without any social cues and tends to suffer from an oscillatory behavior. The Follow The Gap Social Cues (FTG-SC) is the extended follow the gap method with the social cues. The Social Potential Field Social Cues (SPF-SC) is the extended force model with the social cues. The Hybrid Social Cues (Hybrid-SC) method is the is the approach integrating the gap selection strategy into the force model.

7.2 Evaluation of the Social Force

The motivation was to model the social force in the left-side (unless the human is not already too much on the left). To ascertain the same rule, socialistic experiments done by the authors. Cameras were fitted in one of the malls of the city where the people flock in large numbers. The videos were studied to observe the motion of every human. Stress was given to the specific situations where the humans come head-on towards each other. 50 such scenarios were manually mined. In 38 scenarios, the passing by was on the left, while in 12 the passing by was on the right. The right side is primarily

Fig. 8 Socialistic experiments to study passing on the left hypothesis

selected when there is no space in the left and humans' goal is also on right side. The passing on the left therefore has a confidence given by Eq. (30)

$$Confidence = \frac{|left_{pass}|}{|left_{pass}| + |right_{pass}|}$$
$$Confidence = \frac{38}{38+12} = 0.76 \tag{30}$$

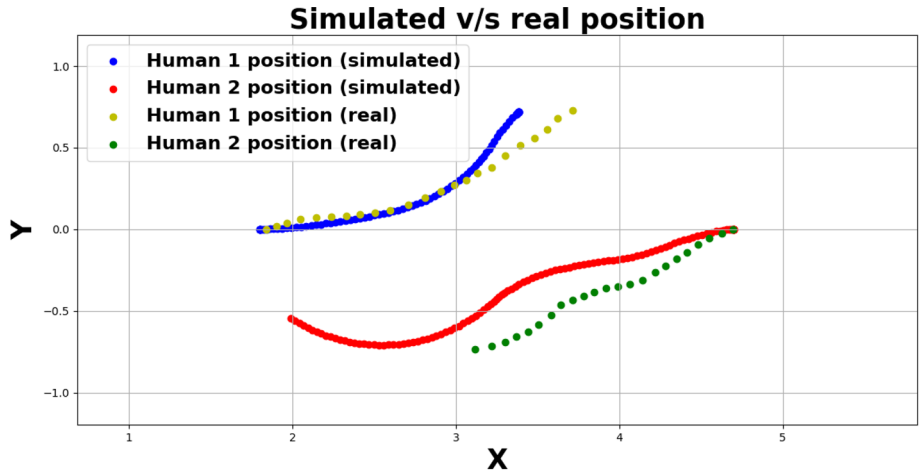
Some snapshots of the passing by are given in Fig. 8. That said, in the Indian scenario the humans usually follow the left side rule in navigation, which is different in other countries and the conventions need to be interchanged.

We further reasoned about the validity of the algorithm where the humans do not stick to the left hand side out of all the observed scenarios. There is always a very brief period where the humans are nearly head-to-head. The human choosing to go on the right is typically very confident and takes a very quick drift. This breaks the uncertainty region in the algorithm about the side of avoidance and the robot shall counter by taking the opposite direction. In our case the robot moves much slower than the people in real life and therefore the robot barely moves in the time when the person dominates shows its strong intent to go from the opposite side.

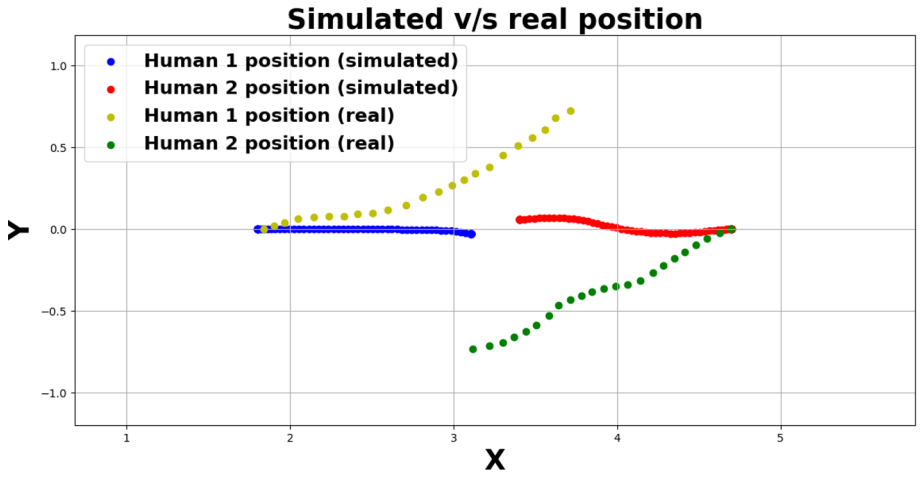
One of the contributions of the paper was to add a social force in the working of the potential field algorithm. In order to socially study the effect of this force, we did a social experiment. Two subjects were asked to walk towards each other naturally and their motion was observed using a 3D lidar. The trajectories are shown in Fig. 9. The simulation was then done using the potential field approach with and without the social force and the results are shown in Fig. 9. Clearly, the social force was better able to imitate the real observed trajectory. Without the social force, the two robots drift apart only when they are very close to each other, while with the social force the robots have in-anticipation given enough space to each other. This proves the effectiveness of the social force.

7.3 Simulations

All the methods have been implemented and tested using the ROS framework on a Linux operating system with configuration 12 GB RAM, CORE I5 processor. MobileSim is the simulator which was used for developing and testing the algo-

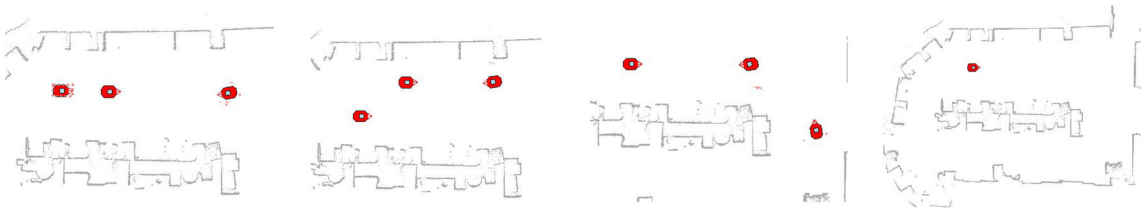


(a) With social force



(b) Without social force

Fig. 9 Effect of the social force on the robot



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3 (d) Scenario 4

Fig. 10 Scenarios used for testing

rithms, before being tested on the real robot. The map used by the simulator was of the Centre of Intelligent Robotics at IIIT Allahabad. The map was made by the Pioneer LX robot using the onboard lidar sensor and a SLAM algorithm. The methods have been tested extensively in various scenarios on simulations. In the simulations, there are multiple instances of the robot that move according by the same algorithm and therefore each acts as a moving human for the other robots to apply the navigational strategies to move in a safe manner. Pioneer LX robot is used in the simulations. The scenarios involve similar situations to that of the real-life navigation of the robots passing each other as well as overtaking scenarios when a robot is overtaken by another. The approaches were tested in the simulations in four different scenarios three of which consisted of at least three dynamic obstacles in the robot's vicinity while the fourth was a static scenario not consisting of dynamic obstacles. In the first scenario, each of the robots have their goals at around 5 metres in front

Table 1 Clearances maintained in meters

Clearance (m)	Scenario 1	Scenario 2	Scenario 3	Static scenario
APF	0.31	0.28	0.32	0.92
FTG-SC	0.35	0.41	0.29	0.84
SPF-SC	0.45	0.43	0.41	0.92
Hybrid-SC	**0.56**	**0.48**	**0.44**	**0.95**

Best results are shown in bold

Table 2 Penalty obtained

Penalty	Scenario 1	Scenario 2	Scenario 3	Static scenario
APF	12.52	17.8	23.6	0
FTG-SC	7.5	10.8	25.4	0.2
SPF-SC	5.72	**6.4**	20.7	0
Hybrid-SC	**0**	7.0	**20.1**	**0**

Best results are shown in bold

Table 3 Time taken in seconds

Time taken (s)	Scenario 1	Scenario 2	Scenario 3	Static scenario
APF	15.10	16.75	31.16	78.3
FTG-SC	15.02	14.85	29.66	84.3
SPF-SC	15.29	15.73	30.86	78.3
Hybrid-SC	15.13	15.43	30.56	83.5

of them. As they move towards their goals, they encounter dynamic obstacles and social cues are used to avoid them. The scenario is shown in Fig. 10a. In the second scenario, the leftmost robot is deliberately placed a little lower to make the motion a bit more complicated. The rightmost robot after having passed the first robot on the left has to go towards its right instead of left as there is no space on the left. The scenario is shown in Fig. 10b. In the third scenario, the robot has multiple sub-goals which is to first move 5 metres straight which is the first sub-goal and then turn right and move 4 metres which is the second sub-goal. Along the way, the dynamic obstacles are encountered, and the social cues are used to avoid them. The scenario is shown in Fig. 10c In the fourth (static) scenario, the robot has to make a round around the entire laboratory. The entire task is broken down into multiple sub goals and the robot moves its current position to its next sub goal. The FSM acts as the global planner and provides the next goal to the robot once the current goal is reached. The entire trajectory is planned such that the robot takes a tour of the entire laboratory. The scenario is shown in Fig. 10d. The minimum clearance maintained by the robot throughout the trial, the time taken to complete the task as well as the penalty incurred by the robot were recorded and presented in the Tables 1, 2 and 3.

The performance of the algorithm is judged by the clearance maintained by the robot and the penalty obtained. Higher the clearance and lower the penalty, the better is the algorithm. From the above observations, it can be inferred that the algorithms accounting for the social cues outperform their non-social counterparts in generating safe trajectories in a pedestrian rich environment. The baseline approach results in the robot moving too close to the human being in many cases and is therefore unsafe. The clearance increases as social cues are introduced into the different methods making the navigation safer. In most of the scenarios, the novel hybrid algorithm Hybrid-SC, which is an integration of the geometric approach into the extended social potential fields, has the best performance as it maintains a higher clearance than the other methods on a consistent basis. The novel hybrid algorithm maintains the highest clearance compared to the other methods in all the scenarios. It also has the least penalty obtained in all but one scenario and hence generates safer trajectories in most cases. The SPF-SC method, which is the social force model extended to account for social forces, has a lesser penalty in one scenario compared to the Hybrid model. The hybrid model being an extension of the force model integrated with the gap selection strategy therefore has proactive characteristics unlike the force field models which are purely reactive. Hence, the novel hybrid method combines the best features of the force model and geometric approaches to outperform the previous model-based methods by generating safer trajectories.

7.4 Hardware Experiments

The algorithm was also implemented on the Pioneer LX robot operating at the Centre of Intelligent Robotics of the institute. The robot is operated using ROS. The robot consists of a lidar sensor that is used to detect the obstacles around as well as for localization. A camera is placed on top of the robot to capture the real time video in order to detect the presence of humans. The face detection algorithm [2] was used to detect the presence of humans in the robot surroundings. As the robot is operating in a known environment, its position at any given time is obtained through localization using a manufacturer supplied library in ROS. The localization, face detection, motion planning and other modules interact as ROS nodes. The scenarios in which the approaches were tested involves the presence of multiple humans. In these experiments, the robot starts from a starting position and is expected to move sequentially towards all sub-goal positions in the environment. As humans are present in this environment, the robot navigates towards its sub-goals by maintaining enough clearance and respecting the social conventions. The scenarios

Table 4 Real world scenario

Pioneer LX	Clearance (m)	Time taken (s)	Penalty
APF	0.34	52.99	59.1
FTG-SC	0.29	51.3	56.7
SPF-SC	0.41	41.3	50
Hybrid-SC	**0.56**	**41.09**	**0**

Best results are shown in bold

usually involve humans walking towards the robot or across the robot causing the social forces from the humans to act on the robot and hence allowing the robot to safely navigate by avoiding collisions. Students working in the laboratory were used as volunteers and were asked to walk through the robot multiple times.

A few snapshots of the run are shown in Fig. 11. The results show the robot first avoids the first human from the correct side, for which the robot had an ample of time and the intention of the person was also clear. However, soon after avoiding the person, the robot had to turn around the corner where it suddenly sees the second person. A natural instinct for the robot while turning would be to be centrally placed, however, encountering the person at the corner, the robot suddenly drifts leftward to give a larger social distance to the person. Thereafter, after another turn, the robot meets a person who's taken a sharp motion towards the robot's right and the robot acknowledges by taking a left. The clearances and angles of the robot during the run are plotted as Fig. 12. A hardware demonstration video of the various methods discussed above can be found at [53].

The comparison with the different approaches is shown in Table 4. Just as observed in the simulations, the Hybrid-SC method outperformed its counterparts even more significantly in the experiments with the real robot. The minimum clearance maintained by the robot from the obstacles was 0.56 meters and more than 10 cm higher than the next best method. The method did not receive any penalty as the minimum distance constraint was never violated. The novel hybrid method has shown to be superior to the previous approaches in both the simulations as well as the real-world experiments on the robot. The hybrid model being an extension of the force model integrated with the gap selection strategy therefore has proactive characteristics unlike the force field models which are purely reactive. Hence, the novel hybrid method combines the best features of the force model and geometric approaches to outperform the previous model-based methods by generating safer trajectories.

7.5 Additional Simulations

The real-world settings take a long time and the requirement of a reasonable number of human actors. Further testing is therefore done using simulations. In simulations multiple instances of robots running under the proposed algorithms is used to analyze the working of the algorithm. The simulations are used to generate many diverse situations that are otherwise seen in the everyday navigation within the laboratory. At the same time, the corner cases of the algorithms including robots ahead of each other are used in the simulations to ensure that the algorithms act appropriately. The simulations enable to place several robots and run the algorithm to see the behavior. The simulations are a good way to ensure that the robots avoid each other with enough clearance and do not block each other or end up in a deadlock, which

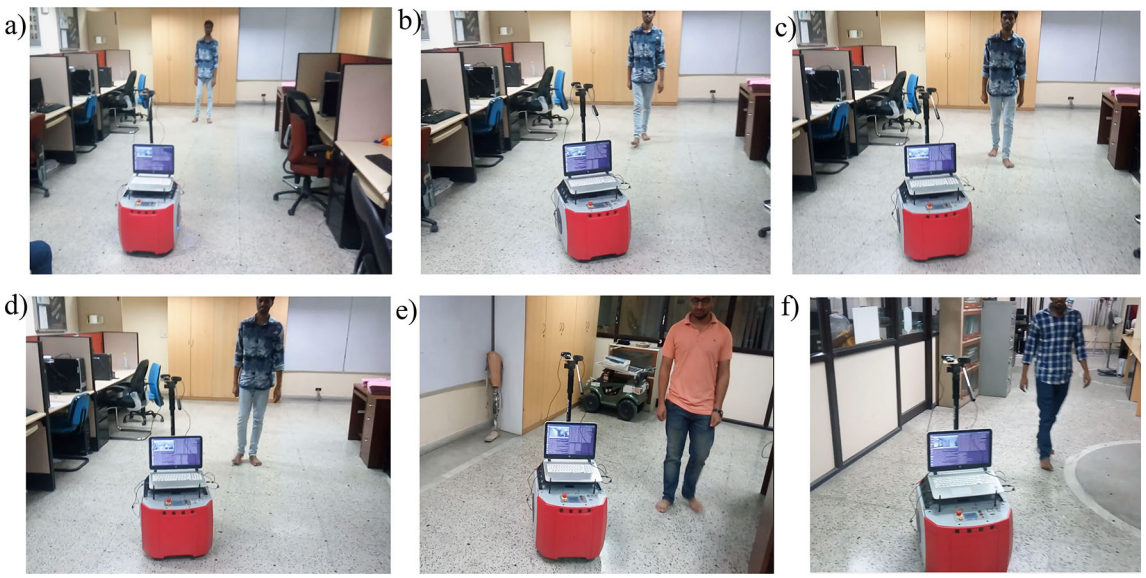


Fig. 11 Execution of the algorithm on the Pioneer LX robot

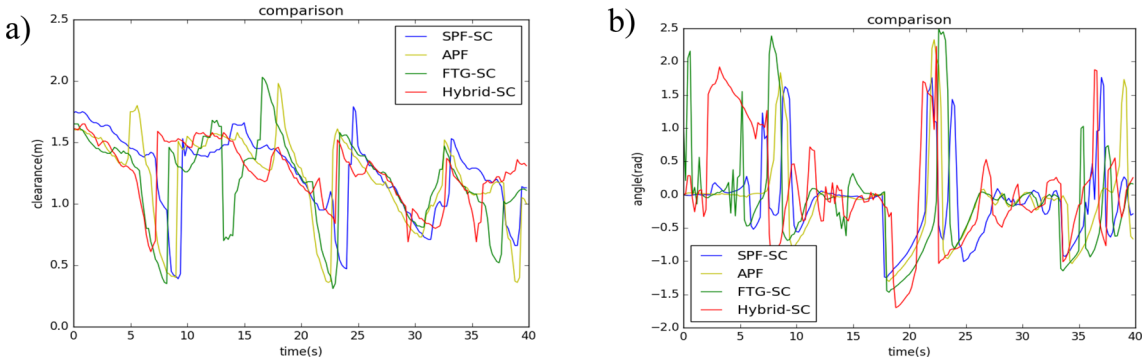


Fig. 12 Metrics on the run with the real robot

Table 5 Simulation results on different scenarios

Scenario	Clearance (m)			Penalty			Time (s)		
	Robot 1	Robot 2	Robot 3	Robot 1	Robot 2	Robot 3	Robot 1	Robot 2	Robot 3
I	0.59	0.60	0.45	0.0	0.0	10.78	27.75	23.57	24.17
II	0.55	0.83	0.42	0.0	0.0	13.89	22.06	22.67	26.90
III	0.33	1.02	0.27	7.07	0.0	10.932	28.41	16.62	14.21
IV	0.5	1.38	0.48	14.8	0.0	17.46	24.78	16.62	30.83
V	0.49	0.51	0.23	11.13	0.0	9.00	22.37	17.22	12.39
VI	0.5	0.55	0.5	10.0	0.0	9.2	16.32	16.92	15.11
VII	0.39	0.33	0.34	14.42	15.27	18.50	22.07	23.27	25.08
VIII	0.5	0.32	0.28	12.6	10.69	10.37	22.36	17.53	13.6
IX	0.42	0.22	0.23	10.90	19.14	61.11	21.76	21.15	54.11
X	0.5	0.5	0.5	8.2	11.0	31.0	22.06	16.93	54.13
XI	0.26	0.29	0.31	17.25	6.88	1.81	22.07	17.23	10.27

is possible for robots. The testing is done on 11 scenarios with 3 robots each. The first 4 scenarios are for overtaking and showing cooperation to overtaking behaviors. The last 7 scenarios test avoidance of robots in different settings. Overtaking may also happen due to speed differences. The variations of speeds and initial inter-robot separation is done in all scenarios. Some of the scenarios are under an adversarial setting where there is a very little distance for the robot to pass by. In scenario I, robot 3 is aggressive and overtakes both the robots. To make matters hard, robot 2 also later attempts to overtake robot 1. In scenario II, robot 3 is to overtake robot 1 and robot 2 which are together ahead. Robot 3 makes the robots drift to the left to allow overtaking. In scenario III, while robot 1 is made to overtake robot 2, the other robots can merely go towards their goal. In scenario IV, the robots are in a line and the challenge is to laterally displace while moving longitudinally towards their goal. In scenario V, while robot 3 overtakes, robot 1 is given a goal opposite to the general convention and robot 2 must hence adjust. Scenario VI features robot 1 and robot 2 with a very high lateral displacement with robot 3 approaching from the other side. The social conventions are simultaneously followed for robot 2 and not followed for robot 3 because of the goal. Robot 3 detects the same and choses the correct side of avoidance, while also carefully balancing the forces. Scenario VII tightens the conditions of Scenario VI by choosing goals with a smaller lateral displacement. Scenario VIII makes robot 2 simultaneously avoid both robot 3 and robot 1 while it is head-on to both of them. In scenario IX, robot 3 must essentially avoid robot 2, while robot 1 makes the lateral separations bound for the two robots. Scenario X makes robot 1 avoid robot 2 from the non-socialistic side, and immediately places robot 3 in front which must also be similarly avoided. Finally, scenario XI makes robot 3 avoid robot 2 easily, which however makes it get head-on with robot 1, which must be quickly avoided. The metrics of the scenarios is shown in Table 5. The initial and the final positions are shown in Fig. 13.

7.6 Analysis

The two basic behaviors presented are robot avoidance when the two robots face each other and cooperative robot overtaking. The behaviors are affected by two factors, the speeds of the robots and the inter-robot distances. Let us therefore analyze the algorithm functioning over these factors. The first behavior is when two robots face each other. Experi-

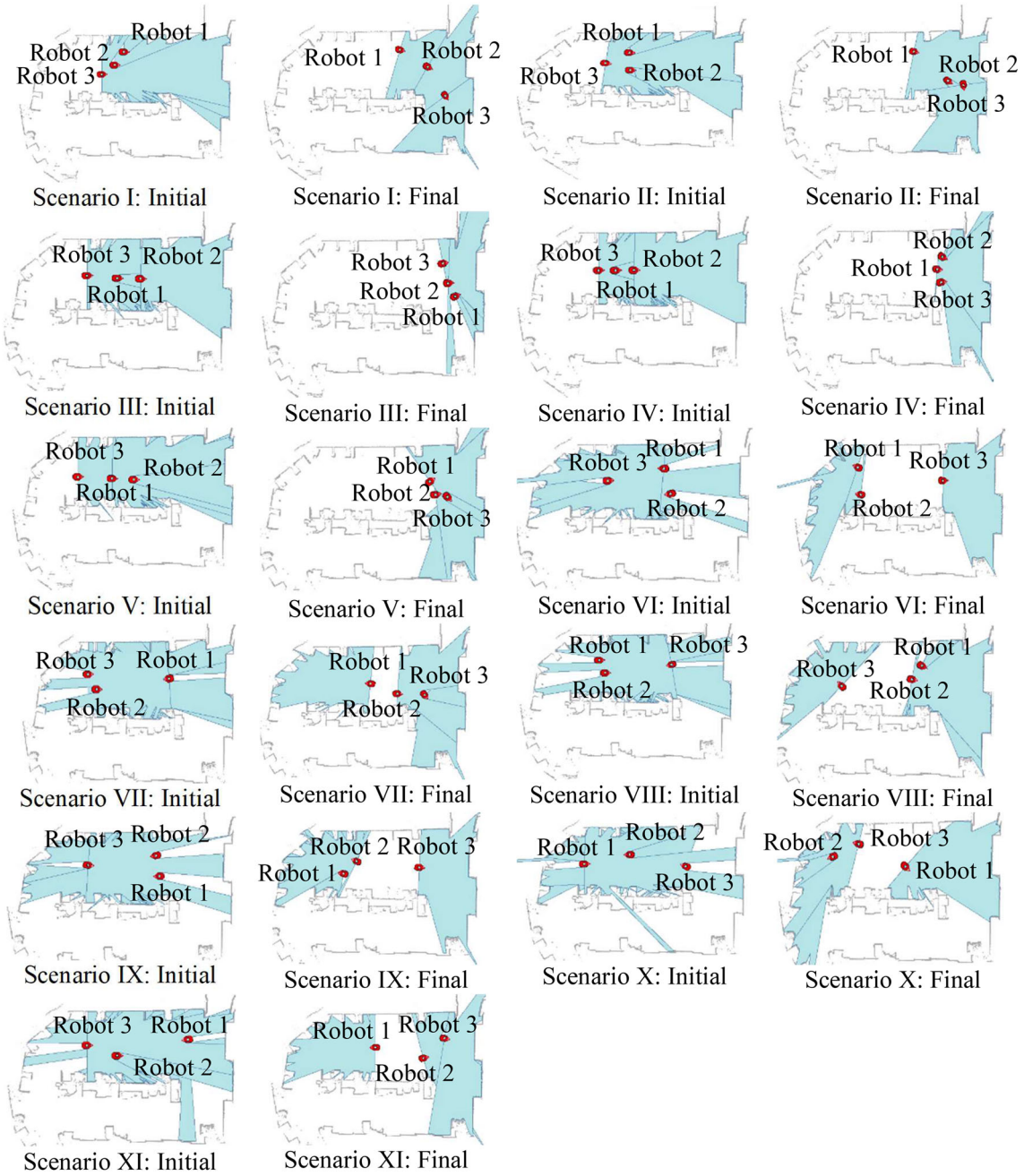


Fig. 13 Simulation results

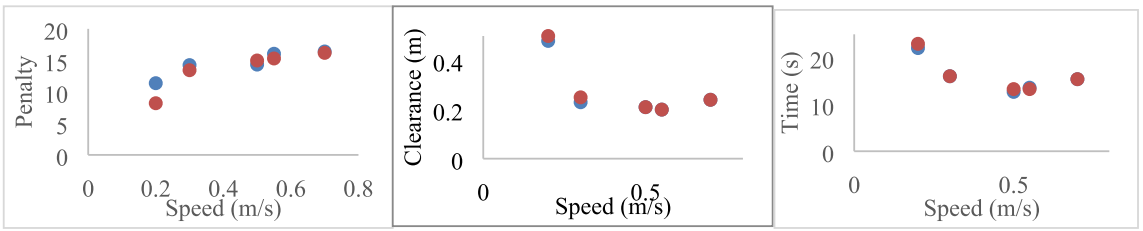


Fig. 14 Analysis of robot avoidance behavior by changing speeds

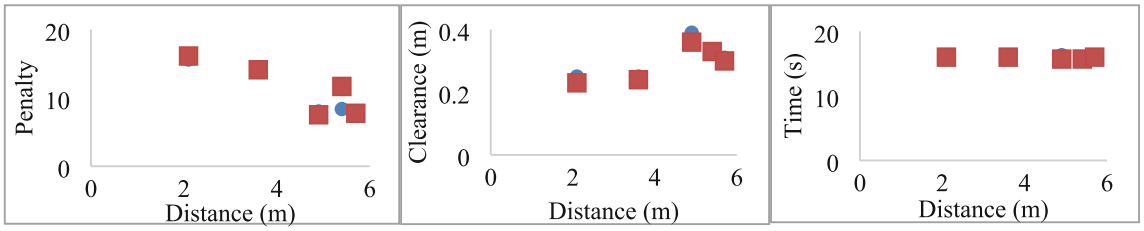


Fig. 15 Analysis of robot avoidance behavior by changing inter-robot distance

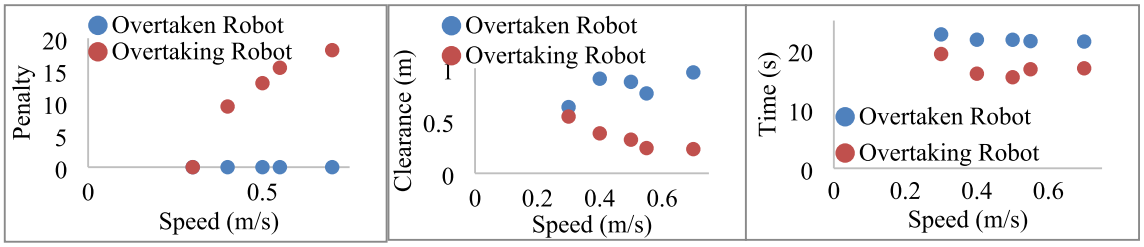


Fig. 16 Analysis of overtaking behavior by changing speeds

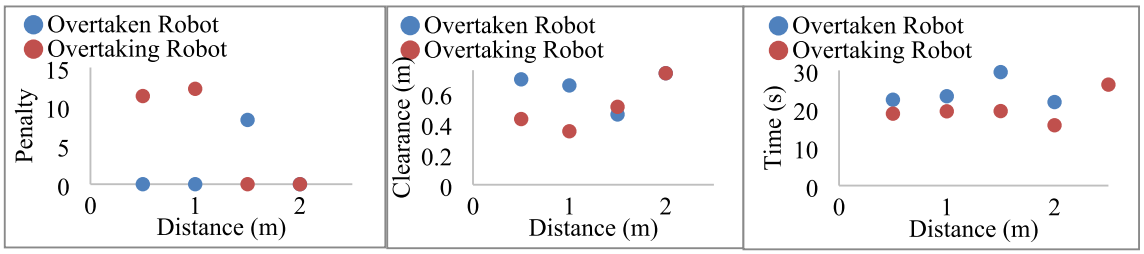


Fig. 17 Analysis of overtaking behavior by changing inter-robot distance

ments are done by varying the speeds and the results are presented in Fig. 14. As the speed increase, the robots get a much smaller time to orient themselves, which reduces the clearance and increases the penalty. The increasing speed, obviously, reduces the travel time. Due to symmetry, the metrics of the two robots are nearly the same, and can be seen as overlapping plots. There is a little deviation due to the stochasticity of the experiments. Similarly, the inter-robot distance is changed and the results are shown in Fig. 15. With a larger inter-robot distance, the robots get more time and hence get larger clearance and smaller penalty. There is no major difference in travel time though.

The other behavior is overtaking. First experiments are done by varying the speed. Higher speed gives a lesser time for adjusting towards a comfortable overtaking distance, which reduces the clearance and increases the penalty for the overtaking robot. The time shows a decreasing trend. The results are shown in Fig. 16. The inter-robot distance is also varied as shown in Fig. 17. If the robots have a large distance, there is more space to adjust to a good overtaking position, which increases the clearance and reduces the penalty. The travel time is reasonable stable. The values for the overtaken robot are due to the limited field of view prior to overtaking and hence the robot misses on measuring the least distance.

8 Conclusion

In this work a social force model as well as a follow the gap method was extended to account for the social cues by enforcing the conventions to be followed. A novel hybrid method was introduced by combining the two methods to retain their respective advantages. The hybrid method combines the reactive nature of the potential fields and the proactive nature of the geometric approach and therefore generates safer trajectories. The method was tested on both the simulations and the robot hardware on multiple scenarios and was shown to outperform the baseline methods by maintaining a larger clearance from the pedestrians.

A typical social robot also interacts with the humans, which was not the current scope of the work. Asking for space to the human and giving human space on demand are social etiquettes of navigation. On the contrary, the social forces and gap selection in this paper was done such that the

robot not only selects the correct side of avoidance, but also voluntarily gives the most possible space to the human. This was done by adding a new orthogonal social force for people in social potential; and biasing the robot for selecting a socially compliant gap and navigating within the gap giving maximum possible space. Our experience with the current robots is that the robots usually move extremely slowly as compared to the humans. So, the humans are always the driving force behind the selection of avoidance strategies, and the robots typically need to be socially compliant to the human dominated obstacle avoidance strategy. Our work was done under the same context and hence the robot never demanded space. The extension of the work for different kinds of robots with different capabilities and for different applications needs to be done. Possibilities for adding an active human–robot-interaction during navigation also needs to be studied. Incorporating the desire, mood and preference by looking at the observed behaviour of the human also needs to be looked in the future. A challenge is that the tracking from a mobile base need to be very accurate as tracking errors can misclassify the preferences, mood or speed parameters.

Acknowledgements This work is supported by the Science and Engineering Research Board, Department of Science and Technology, Government of India vide research grant ECR/2015/000406 and the Indian Institute of Information Technology, Allahabad, India. The authors wish to thank Abhinav Malviya for his help in recording and analysis of the socialistic behaviors used in this study.

Compliance with Ethical Standards

Conflict of interest The authors have no conflict of interest to declare.

References

1. Breazeal C, Dautenhahn K, Kanda T (2016) Social robotics. In: Siciliano B, Khatib O (eds) Springer handbook of robotics. Springer handbooks. Springer, Cham
2. Alač M, Movellan J, Tanaka F (2011) When a robot is social: spatial arrangements and multimodal semiotic engagement in the practice of social robotics. Soc Stud Sci 41(6):893–926
3. Weiss A, Bader M, Vincze M, Hasenhütl G, Moritsch S (2014) Designing a service robot for public space: an action and experiences-approach. In: Proceedings of the 2014 ACM/IEEE international conference on human–robot interaction, 2014
4. Tsuda N, Harimoto S, Saitoh T, Konishi R (2009) Mobile robot with following and returning mode. In: The 18th IEEE international symposium on robot and human interactive communication, Toyama, pp 933–938
5. Shiomi M, Kanda T, Glas DF, Satake S, Ishiguro H, Hagita N (2009) Field trial of networked social robots in a shopping mall. In: 2009 IEEE/RSJ international conference on intelligent robots and systems, St. Louis, MO, pp 2846–2853
6. Bera A, Randhavane T, Prinja R, Manocha D (2017) SocioSense: Robot navigation amongst pedestrians with social and psychological constraints. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), Vancouver, BC, pp 7018–7025
7. Burgoon JK, Jones SB (1976) Toward a theory of personal space expectations and their violations. Hum Commun Res 2(2):131–146
8. Pandey AK, Alami R (2010) A framework towards a socially aware mobile robot motion in human-centered dynamic environment. In: 2010 IEEE/RSJ international conference on intelligent robots and systems, pp 5855–5860
9. Kala R, Warwick K (2013) Motion planning of autonomous vehicles in a non-autonomous vehicle environment without speed lanes. Eng Appl Artif Intell 26(5–6):1588–1601
10. Gopinath D, Jain S, Argall BD (2017) Human-in-the-loop optimization of shared autonomy in assistive robotics. IEEE Robot Autom Lett 2(1):247–254
11. Schirner G, Erdogmus D, Chowdhury K, Padir T (2013) The future of human-in-the-loop cyber-physical systems. Computer 46(1):36–45
12. Ferrer G, Garrell A, Sanfeliu A (2013) Robot companion: a social force based approach with human awareness-navigation in crowded environments. In: IEEE/RSJ international conference on intelligent robots and systems, pp 1688–1694
13. Sisbot EA, Marin-Urias LF, Alami R, Simeon T (2007) A human aware mobile robot motion planner. IEEE Trans Robot 23(5):874–883
14. Helbing D, Molnar P (1995) Social force model for pedestrian dynamics. Phys Rev E 51(5):4282
15. Reif J, Wang H (1995) Social potential fields: a distributed behavioral control for autonomous robots. In: International workshop on algorithmic foundations of robotics, AK Peters, Wellesley, MA, USA, pp 431–459
16. Sezer V, Gokasan M (2012) A novel obstacle avoidance algorithm: follow the gap method. Robot Auton Syst 60(9):1123–1134
17. Kala R (2018) Routing-based navigation of dense mobile robots. Intell Serv Robot 11(1):25–39
18. Murphy R (2000) The hybrid deliberative/reactive paradigm. In: Murphy R (ed) Introduction to AI robotics, 2nd edn. MIT Press, Cambridge
19. Knepper RA, Rus D (2012) Pedestrian-inspired sampling-based multi-robot collision avoidance. In: IEEE international symposium on robot and human interactive communication, pp 94–100
20. van den Berg J, Lin M, Manocha D (2008) Reciprocal velocity obstacles for real-time multi-agent navigation. In: Proceedings of the IEEE international conference on robotics and automation, pp 1928–1935
21. van den Berg, J, Guy, SJ, Lin, M, Manocha, D (2011) Reciprocal *n*-body collision avoidance. In: Pradalier C, Siegwart R, Hirzinger G (eds) Robotics research. Springer tracts in advanced robotics, vol 70. Springer, Berlin. https://doi.org/10.1007/978-3-642-19457-3_1
22. Zhang D, Xie Z, Li P, Yu J, Chen Z (2015) Real-time navigation in dynamic human environments using optimal reciprocal collision avoidance. In: International conference on mechatronics, pp 2232–2237
23. Repiso E, Garrell A, Sanfeliu A (2019) Adaptive side-by-side social robot navigation to approach and interact with people. Int J Soc Robot. https://doi.org/10.1007/s12369-019-00559-2
24. Demir M, Sezer V (2017) Improved follow the gap method for obstacle avoidance. In: IEEE international conference on advanced intelligent mechatronics, pp 1435–1440
25. Ozdemir A, Sezer V (2018) Follow the gap with dynamic window approach. Int J Semant Comput 12(1):43–57
26. Borenstein J, Koren Y (1991) The vector field histogram-fast obstacle avoidance for mobile robots. IEEE Trans Robot Autom 7(3):278–288

27. Ulrich I, Borenstein J (1998) VFH + : reliable obstacle avoidance for fast mobile robots. In: Proceedings of the 1998 IEEE international conference on robotics and automation, pp 1572–1577
28. Ulrich I, Borenstein J (2000) VFH/sup*: local obstacle avoidance with look-ahead verification. In: Proceedings of the 2000 international conference on robotics and automation, pp 2505–2511
29. Khatib O (1985) Real-time obstacle avoidance for manipulators and mobile robots. In: Proceedings of the IEEE international conference on robotics and automation, vol 2, pp 500–505
30. Ferrer G, Garrell A, Sanfeliu A (2013) Social-aware robot navigation in urban environment. In: European conference on mobile robots, pp 331–336
31. Zanlungo F, Ikeda T, Kanda T (2011) Social force model with explicit collision prediction. Europhys Lett 93:68005
32. Shiomi M, Zanlungo F, Hayashi K, Kanda T (2014) 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
33. Takahashi M, Suzuki T, Cinquegrani F, Sorbello R, Pagello E (2009) A mobile robot for transport applications in hospital domain with safe human detection algorithm. In: 2009 IEEE international conference on robotics and biomimetics, pp 1543–1548
34. Görür OC, Erkmen AM (2014) Elastic networks in reshaping human intentions by proactive social robot moves. In: The 23rd IEEE international symposium on robot and human interactive communication, pp 1012–1017
35. Truong XT, Ngo TD (2017) Toward socially aware robot navigation in dynamic and crowded environments: a proactive social motion model. IEEE Trans Autom Sci Eng 14(4):1743–1760
36. Ratsamee P, Mae Y, Ohara K, Takubo T, Arai T (2013) Human–robot collision avoidance using a modified social force model with body pose and face orientation. Int J Humanoid Robot 10(01):1350008
37. Luber M, Spinello L, Silva J, Arras KO (2012) Socially-aware robot navigation: a learning approach. In: 2012 IEEE/RSJ international conference on intelligent robots and systems, pp 902–907
38. Long P, Liu W, Pan J (2017) Deep-learned collision avoidance policy for distributed multiagent navigation. IEEE Robot Autom Lett 2(2):656–663
39. Khare A, Motwani R, Akash S, Patil J, Kala R (2018) Learning the goal seeking behaviour for mobile robots. In: 2018 3rd Asia-Pacific conference on intelligent robot systems, pp 56–60
40. Chen YF, Liu M, Everett M, How JP (2017) Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In: 2017 IEEE international conference on robotics and automation, pp 285–292
41. Everett M, Chen YF, How JP (2018) Motion planning among dynamic, decision-making agents with deep reinforcement learning. In: 2018 IEEE/RSJ international conference on intelligent robots and systems, pp 3052–3059
42. Chen YF, Everett M, Liu M, How JP (2017) Socially aware motion planning with deep reinforcement learning. In: 2017 IEEE/RSJ international conference on intelligent robots and systems, pp 1343–1350
43. Abbeel P, Ng AY (2004) Apprenticeship learning via inverse reinforcement learning. In: Proceedings of the twenty-first international conference on machine learning. ACM
44. Ng AY, Russell SJ (2000) Algorithms for inverse reinforcement learning. In: Proceedings of the international conference on machine learning, p 2
45. Kim B, Pineau J (2015) Socially adaptive path planning in human environments using inverse reinforcement learning. Int J Soc Robot 8(1):51–66
46. Kretzschmar H, Spies M, Sprunk C, Burgard W (2016) Socially compliant mobile robot navigation via inverse reinforcement learning. Int J Robot Res 35(11):1289–1307
47. Truong XT, Ngo TD (2016) Dynamic social zone based mobile robot navigation for human comfortable safety in social environments. Int J Soc Robot 8(5):663–684
48. Paliwal SS, Kala R (2018) Maximum clearance rapid motion planning algorithm. Robotica 36(6):882–903
49. Kruse T, Pandey AK, Alami R, Kirsch A (2013) Human-aware robot navigation: a survey. Robot Auton Syst 61(12):1726–1743
50. Schroff F, Kalenichenko D, Philbin J (2015) Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 815–823
51. Zhang K, Zhang Z, Li Z, Qiao Y (2016) Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process Lett 23(10):1499–1503
52. Quigley M, Gerkey B, Conley K, Faust J, Foote F, Leibs J, Berger E, Wheeler R, Ng A (2009) ROS: an open-source robot operating system. In: ICRA workshop on open source software, vol 3, no 3.2, p 5
53. Reddy AK, Malviya V, Kala R. Video results. https://www.youtube.com/watch?v=JZt-798I7cw&list=PLWKZs3S5723nUEENFte7pvHRO49YIm8Tj

Arun Kumar Reddy received his masters' and bachelors' degrees in Information Technology with a specialisation in Robotics from the Indian Institute of Information Technology, Allahabad (Prayagraj) in 2019. He is currently working as a software developer and at Flipkart for the past year as a part of the core apps team. His interests include application development, Robotics and Artificial Intelligence. His hobbies include learning new skills and technologies, watching football and reading books.

Vaibhav Malviya received the M.Tech degree in Computer Science & Engineering from Jaypee Institute of Information Technology, Noida, India in 2015 and he is currently pursuing Ph.D from the Indian Institute of Information Technology Allahabad (Prayagraj), India. His research interests are vision application for robotics and robot motion planning.

Rahul Kala received the B.Tech. and M.Tech. degrees in Information Technology from the Indian Institute of Information Technology and Management, Gwalior, India in 2010. He received his Ph.D. degree in cybernetics from the University of Reading, UK in 2013. He is currently working as an Assistant Professor in the Indian Institute of Information Technology, Allahabad, India in the Centre for Intelligent Robotics. He is the author of four books and over 100 papers. He has numerous funded projects from Government agencies and industry to his name. He is a recipient of the Early Career Research Grant from the Department of Science and Technology, Government of India. He is a recipient of the Best PhD dissertation award from the IEEE Intelligent Transportation Systems Society; and a scholarship under the Commonwealth Scholarship and Fellowship Program from the UK Government.