



# Waypoint based path planner for socially aware robot navigation

Hasan Kivrak<sup>1,2</sup> · Furkan Cakmak<sup>3</sup> · Hatice Kose<sup>2</sup> · Sirma Yavuz<sup>3</sup>

Received: 21 March 2021 / Revised: 6 November 2021 / Accepted: 13 November 2021 / Published online: 8 January 2022  
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

Social navigation is beneficial for mobile robots in human inhabited areas. In this paper, we focus on smooth path tracking and handling disruptions during plan execution in social navigation. We extended the social force model (SFM)-based local planner to achieve smooth and effective social path following. The SFM-based local motion planner is used with the A\* global planner, to avoid getting stuck in local minima, while incorporating social zones for human comfort. It is aimed at providing smooth path following and reducing the number of unnecessary re-plannings in evolving situations and a waypoint selection algorithm is proposed. The whole plan is not directly assigned to the robot since the global path has too many grid nodes and it is not possible to follow the path easily in such a dynamic and uncertain environment inhabited by humans. Therefore, the extracted waypoints by the proposed waypoint selection algorithm are incrementally sent to the robot for smooth and legible robot navigation behavior. A corridor like scenario is tested in a simulated environment for the evaluation of the system and the results demonstrated that the proposed method can create paths that respect people's social space while also eliminating unnecessary replanning and providing that plans are carried out smoothly. The study presented an improvement in the number of replannings, path execution time, path length, and path smoothness of 90.4%, 53.7%, 8.3%, 55, 2%, respectively.

**Keywords** Social navigation · Mobile robots · Human–robot interaction · Path planning

## 1 Introduction

Robots are becoming a part of our lives and we expect robots to act in a way to avoid interference with our safety and social being. Robots which are employed in human inhabited areas such as malls or hospitals should benefit from a navigation approach that is built upon those principles. This navigation is more than mere avoidance and requires legible robot motion so that rational agents as humans should under-

stand and predict the robot motion (eliminate uncertainty in robot behavior) to adapt their motions accordingly. Furthermore, especially in times of pandemics, robots obeying social distance during the navigation process among people is a positive step for the acceptance of robots and increases the physical and social comfort of the bystanders.

The proposed approach will be used as a part of the novel Roborehab social robotic system involving an affective Pepper humanoid robot and a sensory setup which is designed and developed for the assistance of children with hearing impairments in hospitals [8,18]. This paper focuses on the social navigation of this assistive mobile robot among children, families, and clinicians in hospitals where the robot is expected to generate its own path to a desired location in these human–robot interactive environments, considering the social distance requirements.

✉ Hasan Kivrak  
kivrakh@karabuk.edu.tr

Furkan Cakmak  
furkan@yildiz.edu.tr

Hatice Kose  
hatice.kose@itu.edu.tr

Sirma Yavuz  
sirma@yildiz.edu.tr

<sup>1</sup> Present Address: Department of Computer Engineering, Karabuk University, Karabuk, Turkey

<sup>2</sup> Department of Artificial Intelligence and Data Engineering, Istanbul Technical University, Istanbul, Turkey

<sup>3</sup> Department of Computer Engineering, Yildiz Technical University, Istanbul, Turkey

## 2 Related work

Autonomous robot navigation refers to the process of the finding and execution of a collision free trajectory from the start pose of the robot to the intended target pose. In order to

complete specific tasks in the environment, maps with obstacles that are known to the robot in advance, i.e., grid based maps, are utilized. A\*, Dijkstra and RRT\* are well known search algorithms that address the problem of path planning. Learning and optimization based algorithms, such as fuzzy neural networks with particle swarm optimization also attracted scholars attention to solve path planning problem [9]. A navigation algorithm based on the Particle Filter and SLAM is proposed to be used in the dynamic environments [19]. The aforementioned methods for solving the navigation problems calculate collision-free paths, but do not specifically focus on generating a social path that takes social factors like physiological comfort and proxemics into account.

On the other hand, socially aware navigation approach is a new active research field combining perception and social intelligence. The primary motivation of these approaches is to increase the safety and psychological comfort in human–robot interactive social environments as much as possible. Proxemics theory-based methods attach space or costs to map cells while planning a path around the center of humans or interactions for modeling the social context. This method modifies the robot's costmaps to reflect social behaviors and social zones. The studies in [5,16] modeled proxemics into a robot's local costmap. Their socially aware costmaps follow two dimensional Gaussian distributions or constant values of a different mixture of social constraints depending on the spatial context, e.g., adding more costs in the direction of human orientation. However, this requires fine-tuning of the costmap parameters because one or more dynamic and static people around the robot may block the possible paths of the robot. The robot might freeze, and can not move although there is available space for action in a cluttered and narrow hallway. For this reason, Reference [10] modified the social costmap by using softer constraints on the planned paths (a linear cost function) to allow the robot to enter places, if needed, where it normally would not. Furthermore, several researchers have introduced variants of the dynamic window approach (DWA) which expand the optimization functions for social robot navigation [1]. Another work applies a prediction step to the DWA algorithm and then uses an adaptive neuro-fuzzy inference system to adjust DWA weights [15].

For online motion planning, potential fields are a common approach for static environments. This approach was adopted as a traditional social force model (SFM) to describe the motion of pedestrians in crowded escape scenarios. The model is then successfully employed as a local planner in a corridor like environment [7] and urban human scenarios [2]. However, traditional SFM fails to mimic for a diverse range of social interactions which are present in human environments. Because SFM has a flexible nature by its mathematical formulation, it can be extended with social signals and cues [12] and walking side-by-side [14] which affect social interactions. The problem of local minima should also be addressed

during the implementation of the SFM approach. Therefore, the motion planning problem may be separated into global planning and path following algorithm to execute the produced global plan as two-layered design. Obstacle avoidance strategies can be incorporated into motion planning since using only global planners in a dynamic environment might be slow. Furthermore, once the initial path is obtained, it may not always be expected to stay the same and be easy to follow the path due to the nature of navigation in evolving dynamic environments.

In the light of the above consideration, in this study, we extend our prior online path following approach [6] to include further flexibility in dynamic human environments that may have one or more local minima conditions. This is achieved by adding a waypoint selection algorithm considering human behaviour patterns. By means of this novel extension, we expect to provide smooth motion, avoid getting stuck at local minima, and minimize disturbance in the original plan (e.g., the number of unnecessary replanning, travel time), while retaining the safety and comfort of the people in interaction to the robot. Furthermore, we will use the social individual index (SII) for the evaluation of this system in terms of social comfort. SII is a parameter which is used to measure human physical and psychological safety and comfort during robot navigation [17].

This system is designed and developed to be used with children with hearing impairments in hospitals and audiology departments. Therefore, safety and social comfort are vital as well as the smoothness of the motion and cost effectiveness (in terms of time and path length) of the motion of the robot to be acceptable and useful on behalf of the users in this scenario.

### 3 Extended SFM-based local planner

SFM can be regarded as a kind of potential field path planning in which mathematical functions are followed in the configuration space (think of a vector field over the space). The SFM-based motion planner is computationally light, which is appropriate in uncertain dynamic environments for steering the robot during its navigation. According to this model, the agent behavior is affected by attractive and repulsive forces for acceleration, deceleration and directional changes. The idea behind the model makes it a good candidate for local path planning and is expected to generate more human-like trajectories for the robot. That enables a robot to imitate the comprehensibility of the inner dynamics of human motion efficiently dependent on its motion constraints.

The attractive force is the force that pulls the agent towards the desired destination. The force is velocity dependent and stated as a composition of the desired speed, desired direction, and current velocity using Newton's motion law. The

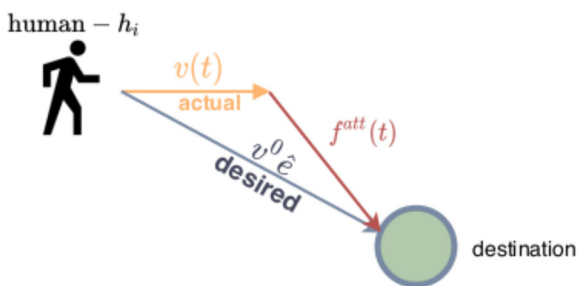


Fig. 1 Illustration of the attractive force to the destination

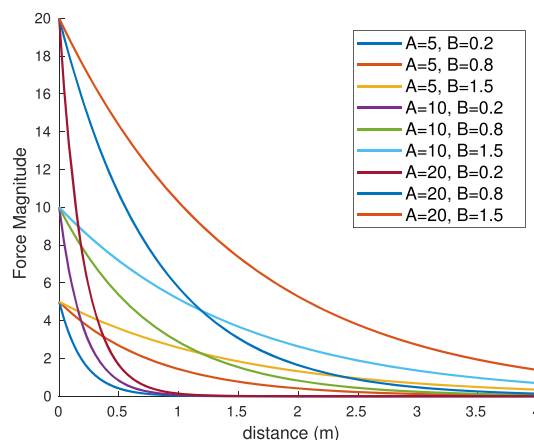
desired speed is effected by the crowd of the place or in a rush. The desired velocity,  $\mathbf{v}^0 = v^0 \cdot \hat{\mathbf{e}}$ , is equal to the desired speed  $v^0$  multiplied by the desired direction  $\hat{\mathbf{e}}$ . The desired direction is the shortest path unit vector, which is the direct path from our current pose. When moving, the agent moves ideally at the desired speed. When the desired speed is currently slowing down due to disturbing factors, it tends to adjust its actual velocity,  $\mathbf{v}$ , in order to reach the desired velocity at the relaxation time,  $k^{-1}$ . As a result, the attractive force is visualized and expressed in both Fig. 1 and Eq. 1.

$$\mathbf{f}^{att}(t) = k \cdot (\mathbf{v}^0(t) - \mathbf{v}(t)). \tag{1}$$

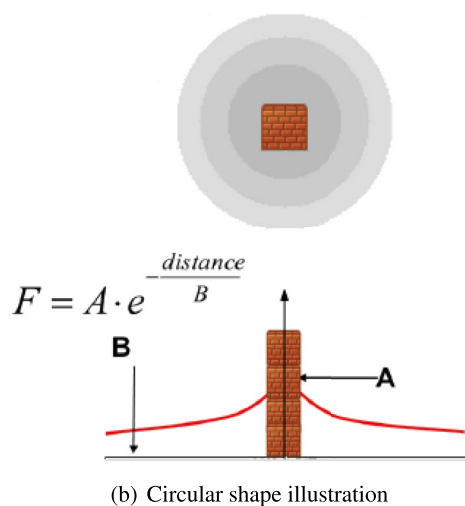
The motion is deviated by the repulsion forces of other people and obstacles around. The force that pushes people or obstacles away in order to maintain a safe distance is called a repulsive or interaction force. Deviations from the shortest path are the result of these types of forces. Repulsive forces considered to be circular shape is expressed in terms of the relative Euclidean distance between the obstacle and the robot as defined in Eq. 2. It employs the Euclidean distance vector  $\mathbf{d}_{ij} = \mathbf{p}_j - \mathbf{p}_i$  decays with exponential, indicating from the position  $i, \mathbf{p}_i$  to the position  $j, \mathbf{p}_j$ . When approaching an obstacle, the force is strong, and when away from an object, the force magnitude is weaker and even will not deviate from the current motion of the agent. An illustration of the circular shape repulsive function of relative distance can be seen in Fig. 2b.

$$\mathbf{f}_{ij}^{rep}(t) = \sum_{j \in Q_o} A_o \cdot e^{-\mathbf{d}_{ij}/B_o} \cdot \hat{\mathbf{d}}_{ij}. \tag{2}$$

$A_o$  is the repulsive force strength or intensity, while  $B_o$  is the interaction range. The range factor controls how quickly the force decreases based on the distance. If the value of the range parameter is big, then the force will slowly decrease and vice versa. The exponentially decreasing repulsion function with various parameter values is seen in Fig. 2a.  $Q$  is the entity type [obstacle ( $Q_o$ ) or human ( $Q_h$ )].  $\hat{\mathbf{d}}_{ij} = \mathbf{d}_{ij}/\|\mathbf{d}_{ij}\|$  is the normalized distance vector between features. The total



(a) Exponential decaying function with different values



(b) Circular shape illustration

Fig. 2 Illustration of the a exponential decaying and b circular shape repulsive function of relative distance with different values

repulsive force is calculated as the sum of all the repulsive forces exerted by each entity.

The superposition (sum) of attractive and repulsive forces guides the robot toward the goal while simultaneously avoiding each obstacle and increasing the physical and physiological safety of the human (Eq. 3).

$$\mathbf{f}^{total}(t) = \mathbf{f}^{att}(t) + \mathbf{f}^{rep}(t). \tag{3}$$

At every point in time, the robot looks at the resultant total force at that point and applies a control law to determine the direction of travel and speed. It is required to turn the model output in the Cartesian coordinates into kinematic robot motion in polar coordinates (linear and angular velocities). More details are given in Sect. 3.1.4. For information about the calibration process of SFM parameters considering the size of the robot and proxemic zones, please refer to [6].

### 3.1 Extensions

Reactive planners are found to be late in taking action because they do not take possible future collisions into account in dynamic human environments. Conventional SFM is designed for reactive collision avoidance by keeping a certain distance from humans and obstacles. Proxemic theory is reflected implicitly by the repulsive force generated by humans in their social context so that it does not show proactive behavior in passing and crossing scenarios. According to [20], none of the elliptical-based force specifications are intended to predict pedestrian behavior and future interactions. The extensions that are added to the traditional SFM method are explained in the following subsections.

#### 3.1.1 Passing and crossing behavior

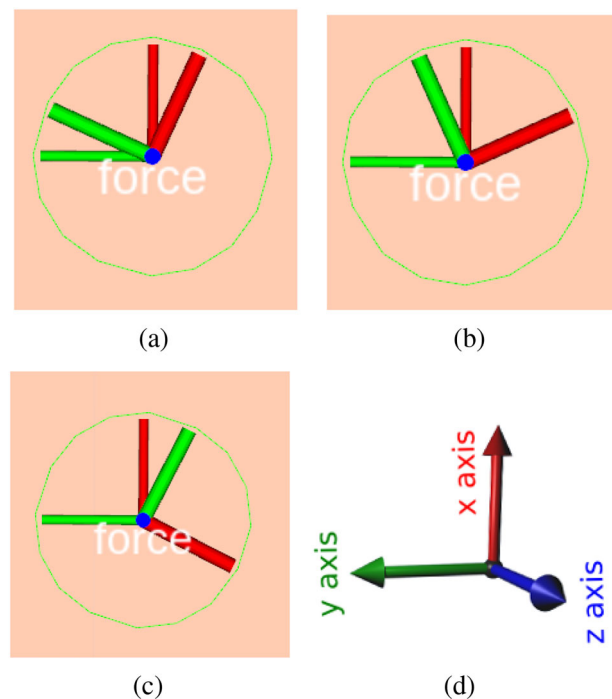
The proactive behavior in the motion model is beneficial in navigating alongside people. Passing and crossing behavior can be done by incorporating the collision prediction scheme that makes the robot have proactive motion depends on relative positions and speed. We implement the collision prediction force introduced in [20] to calculate the repulsive forces from humans. The collision prediction force applies a circular force at a specific angle to the potential conflict point between the robot and the person as they move towards each other. A collision prediction-based passing force has an internal dynamic that actively attempts to keep humans at bay in passing scenarios. Similarly, it allows the agent nearest to the crossing point to accelerate while the other slows down.

#### 3.1.2 Instant turns and oscillations

The other problem with algorithms that use potential field ideas is instant turns and oscillations that could happen because of the instant changes in force size and discontinuity at some points. These may reduce the robot's motion understandability and acceptance. To suppress oscillations and sudden changes when the robot moves in any direction, a kind of interpolation or smoothing is applied to the consecutive time stamp forces. That is, the forces are decreased or increased by a specific step size when the consecutive time-stamped force magnitudes are bigger than a certain threshold. That way, we impose continuity on the steering.

#### 3.1.3 Identifying force frame

A structure that has fixed orientation as the global orientation and movable origin based on the robot's pose is required to be able to evaluate and perform vector operation of forces. It is handled by publishing a force frame transformation as seen in Fig. 3.



**Fig. 3** Force global frame is not changing while the robot is turning right

#### 3.1.4 Robot controller

Due to the inherent constraints on a wheeled platform, the linear ( $v_r$ ) and angular ( $\omega_r$ ) velocities of non-holonomic robots govern their locomotion. To this end, the output vector of the SFM algorithm is required to be transformed into velocity command in polar coordinates for the robot controller. There are several proposed methods for making the robot compatible with the SFM approach. One approach exploits the algebraic description of vectors. The total force vector,  $f^{total} = f_{r\parallel\theta} + f_{r\perp\theta}$  can be broken into x and y components in a two-dimensional coordinate system. The amplitude of the projected component on the heading direction  $f_{r\parallel\theta}$  and the orthogonal component  $f_{r\perp\theta}$  can be considered to contribute into translational acceleration  $a_v$  and rotation acceleration  $a_\omega$ , respectively. Hence, the robot rotation acceleration can be computed in two different ways as in Eq. 4 [3] and Eq. 5 [13].

$$a_\omega = r \times f_{r\perp\theta} + k\omega, \quad (4)$$

where  $r$  is the robot's vector radius towards  $\theta$  and  $k$  is a factor.

$$a_\omega = k_p \cdot \Delta\alpha + k_d \cdot (-\omega), \quad (5)$$

where  $\Delta\alpha$  is the angle between the total force vector and the robot orientation and  $k$  is a factor. Linear and angular velocities is then computed using the following equations.

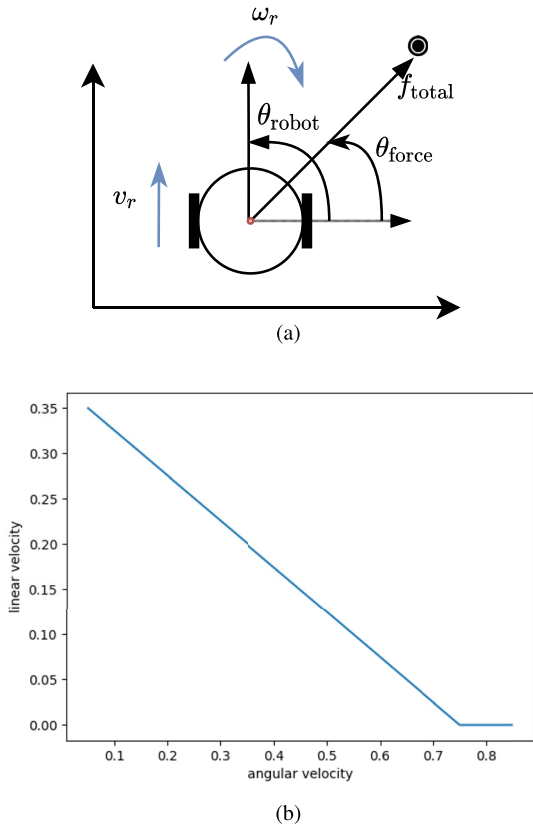


Fig. 4 SFM-based robot controller ( $v, \omega$ )

$$v_r = v_r + a_v \cdot \Delta t, \tag{6}$$

$$\omega_r = \omega_r + a_\omega \cdot \Delta t. \tag{7}$$

In our implementation, we determine the resultant force as an indicator of the particular direction the robot should follow in the next period.

The degrees of  $(\theta_{force} - \theta_{robot})$  pointing in the direction of the  $f^{total}$  force are sent to a steering input called  $\omega_r$  (Fig. 4a). Linear speed is then determined under the assumption that the robot needs less speed at high steering, as shown in Fig. 4b. Note that the yaw angle is the rotation on the robot’s z axis and can range from  $\theta \in [-\pi, \pi]$ . When the robot is on a surface, the yaw angle difference must be scaled to a  $[-\pi, \pi]$  range by either adding or subtracting  $2\pi$  based on the value of  $\theta$  as expressed in Eq. 8.

$$\theta = \begin{cases} \theta - (2\pi) & \text{if } \pi < \theta < 2\pi, \\ \theta + (2\pi) & \text{if } -2\pi < \theta < -\pi. \end{cases} \tag{8}$$

### 3.1.5 Avoiding local minima

There is a known issue with algorithms based on artificial potential areas that could be stuck in local minima, as stated in [4]. This happens when the direction of the robot velocity,

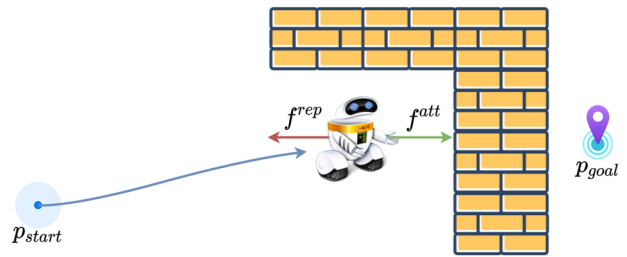


Fig. 5 Local minima condition, where the sum of the attractive and repulsive forces are balanced out

the obstacle, and the target position are on the same straight line and the obstacle is positioned between the robot center and the target location. The robot is attracted towards the target while at the same time pushed away from an obstacle, as shown in Fig. 5. Because local minima lead to a situation in which the robot gets stuck between the target location and the obstacle where the repulsive force balances out the attractive force. This defect makes the method incomplete and the robot never reaches its goal.

To overcome this problem, we incorporate the high-level global planner to find a valid path between the starting point and goal point. The computed global path created by a global planner is transferred to the SFM-based local planner to execute the plan.

### 3.1.6 Waypoints (social subgoals) selection algorithm

During plan execution, the initial plan tends to change frequently in dynamic or uncertain environments. Therefore, re-planning is needed. Frequent changes in the plan can be costly and yield to non-smooth motion. Moreover, the whole global plan consists of too many grid nodes and it is infeasible to have a smooth motion planning algorithm in such environments. To avoid unnecessary replanning and provide smooth motion control, we propose a waypoint selection algorithm. Human social zones (social costmaps) are considered to respond to disruptions. The waypoints of the global path are extracted by pruning parts of the global path and then they are incrementally assigned as waypoints to the robot’s path planner. The method for computing the waypoints is given in the Algorithm 1. Since the key points can be different for each of the interaction, an algorithm is developed based on the angle between the vectors of the node. The definition of the algorithm steps for selecting the waypoints from global path are as follows:

First of all, the global plan  $\mathcal{P} = \{p_1, p_2, p_3, \dots, p_n\}$  that consists of all the grid nodes towards the goal position is obtained, where  $p_n$  is the goal location as the last point. Then, an empty waypoint list  $\mathcal{K}$  is allocated. Following that, we go through all of the nodes and score each point  $i$  based on the metric of angle between vectors. We keep  $p_i$  as the current

point,  $p_{i-1}$  as the previous point, and  $p_{i+1}$  as the next point, and compute  $p_i$ 's predecessor vector  $\vec{v}_1$  and  $p_i$ 's successor vector  $\vec{v}_2$ . The angle  $\theta$  between  $\vec{v}_1$  and  $\vec{v}_2$  is calculated by taking the dot product of vectors to determine whether or not  $p_i$  is a waypoint. If  $\theta$  is less than a certain value, then set  $p_i$  as the next waypoint  $k_i$ , and add it to  $\mathcal{K}$  before moving on to next point. When all nodes have been traversed, i.e., all waypoints have been determined, then set the last node, which is the goal point, as the last waypoint list  $\mathcal{K}$ . Finally, the algorithm returns all the waypoints denoted as  $\mathcal{K} = \{k_1, k_2, k_3, \dots, k_m\}$  along the global path. Examples where waypoints are found on the global plan while incorporating human social cost-maps are illustrated in Fig. 7.

---

**Algorithm 1:** Waypoints (social subgoals) selection algorithm

---

**Input:** Global Path,  $\mathcal{P} = \{p_1, p_2, p_3, \dots, p_n\}$   
**Output:** Waypoints,  $\mathcal{K} = \{k_1, k_2, k_3, \dots, k_m\}$   
 $\mathcal{K} = \emptyset$   
**foreach**  $p_i \in \mathcal{P}$  **do**  
  calculate the angle value of any node  $p_i$ :  
   $\vec{v}_1 = p_i - p_{i-1}$   
   $\vec{v}_2 = p_{i+1} - p_i$   
   $\cos(\theta) = (\vec{v}_1 \cdot \vec{v}_2) / (\|\vec{v}_1\| \|\vec{v}_2\|)$   
  **if**  $\theta < \text{threshold}$  **then**  
  | erase  $p_i$  from the global plan  $\mathcal{P}$   
  **else**  
  | add  $p_i$  to waypoints  $\mathcal{K}$   
  **end**  
**end**  
**return** waypoints  $\mathcal{K}$

---

## 4 Simulation

ROS is the de facto standard in research robotics and offers us the ability to use multiple platforms, languages and incorporate standard solutions to robot problems.

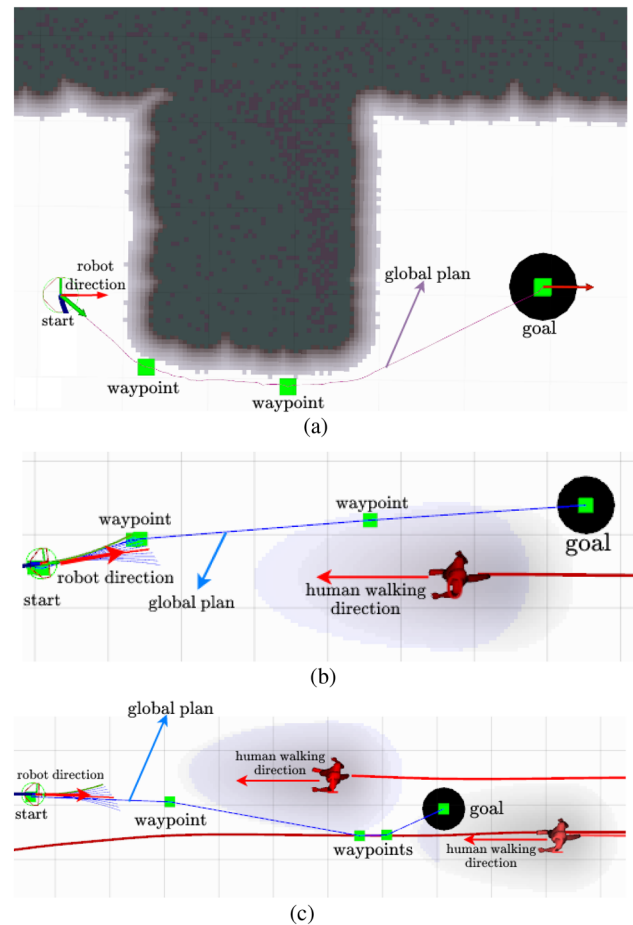
### 4.1 Integration of ROS navigation stack with Pedsim simulator

Pedsim<sup>1</sup> is developed as a ROS package using publicly available Pedsim library<sup>2</sup> based on the SFM. It is an environmental representation tool that can be used on Rviz. This package is extended to integrate move\_base<sup>3</sup> navigation stack with our robot. move\_base is a ROS meta-package that contains a navigation framework using a costmap based approach to planning on top of a map. It makes use of a global planner that does a shortest-path search (using A\* or Dijkstra's algorithm) along the grid of its currently known map (and replans

<sup>1</sup> [https://github.com/srl-freiburg/pedsim\\_ros](https://github.com/srl-freiburg/pedsim_ros).

<sup>2</sup> <https://sourceforge.net/projects/pedsim/>.

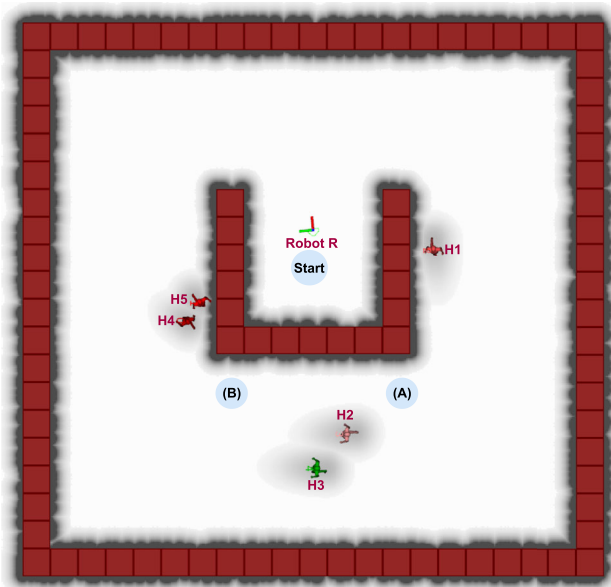
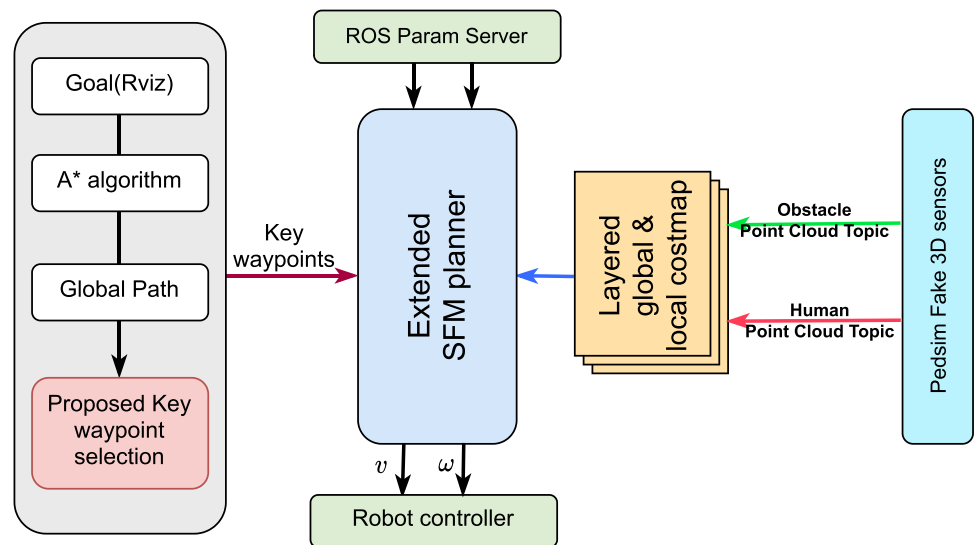
<sup>3</sup> [http://wiki.ros.org/move\\_base](http://wiki.ros.org/move_base).



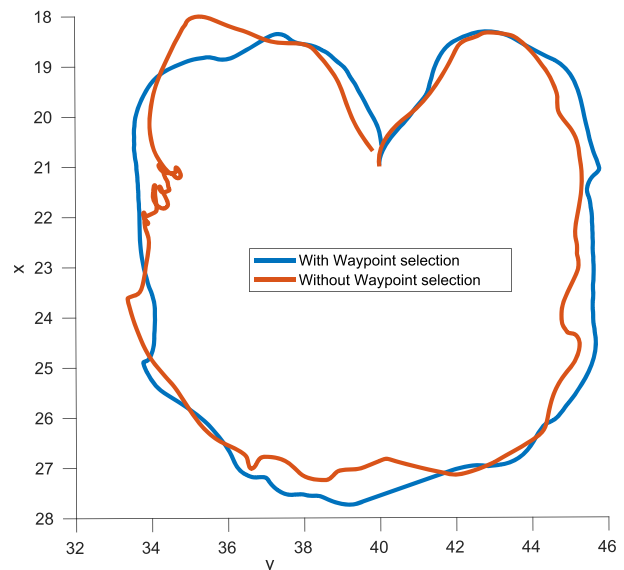
**Fig. 6** Software architecture overview of the waypoint based extended SFM planner

when the local planner cannot succeed in following the path and avoiding dynamic obstacles). The navigation stack uses two different cost-maps that store information about obstacles in the environment. The first one is called “global\_costmap” which contains the whole obstacle map of the environment. The second one is called “local\_costmap” which contains the local area of the map with sensor measurements and is used to avoid obstacles. The separate configuration files with the common configuration (e.g., obstacle thicknesses, colors, obstacle threshold values) file are created. Furthermore, the ROS costmap library provides a representation of a layered cost map and allows us to work with different layers. Such as static maps, obstacle layer, inflation layer, and other user-defined layers (for example, socially compliant cost maps). One can create a human-aware layer for proxemic cost maps such that it subscribes to the location of people and modifies the cost maps by adding Gaussian costs around the detected static or dynamic people. Therefore, the robot navigates between people without collision and respects their personal or social areas (proxemic concerns) [11].

**Fig. 7** Process of finding or selecting and extracting waypoints on the global plan (solving local minima problem smoothly)



(a) Experiment scenario



(b) Trajectories of the robot generated by the w/out waypoint approaches.

**Fig. 8** The corridor-like dynamic simulation environment consists of walls and moving humans where the robot can get stuck at the local minima conditions. The robot R navigates consecutively from the start

point (Start) through points (A) and (B), and then goes back to the start point while avoiding encountering dynamic humans H1–5 and obstacles in the environment

With the help of pedsim\_sensor package<sup>4</sup> the simulator provides the point cloud of “dynamic human obstacles” and “static obstacles” in world frame. Social navigation cost-map layers<sup>5</sup> are added for humans as layered cost maps which increase the map in the direction of human motion. The layer subscribes to the position of people and adds a cost map with a Gaussian distribution around the people. Eventually, the navigation stack on Pedsim simulator using the sensory

data and cost map configurations are developed on ROS. The major component of modules that is used in the experiments are depicted in Fig. 6.

### 4.2 Simulation experiments

Simulation experiments are implemented to demonstrate the performance of proposed approach for social navigation on ROS Melodic on Ubuntu 18.04 (Bionic Beaver). We simulate a corridor-like dynamic environment with possibilities where the robot can get stuck at the local minima condition on

<sup>4</sup> <https://tinyurl.com/h5hrj2y4>.

<sup>5</sup> [http://wiki.ros.org/social\\_navigation\\_layers](http://wiki.ros.org/social_navigation_layers).

**Table 1** Comparison of w/out waypoint selection approaches on characteristic of path properties

Evaluation metrics	Without waypoint approach	Proposed waypoint approach
Path execution time (s)	441	204
Path length (m)	43.652	40.030
#replanning	21	2
Total heading change (°)	5094	2282

Pedsim. We compare our approach to the traditional method (without proposed algorithm) in the scenario illustrated in Fig. 8a. In each case, the robot navigates sequentially from the start point (Start) through points (A)–(B), and then goes back to the start point with the aim of socially avoid dynamic humans and obstacles in the environment.

The dimensions of the area created in the Pedsim simulation environment are 20 m × 21 m. The evaluation metrics used to assess the efficiency of the proposed idea are path execution time, path length, number of replanning, and cumulative heading changes. Heading changes along the trajectory provides an easy way to check for path smoothness. Furthermore, human comfortable safety indices are suggested to measure socially appropriate behaviors in mobile robots. The SII is used to measure human physical and psychological safety and comfort during robot navigation [17].

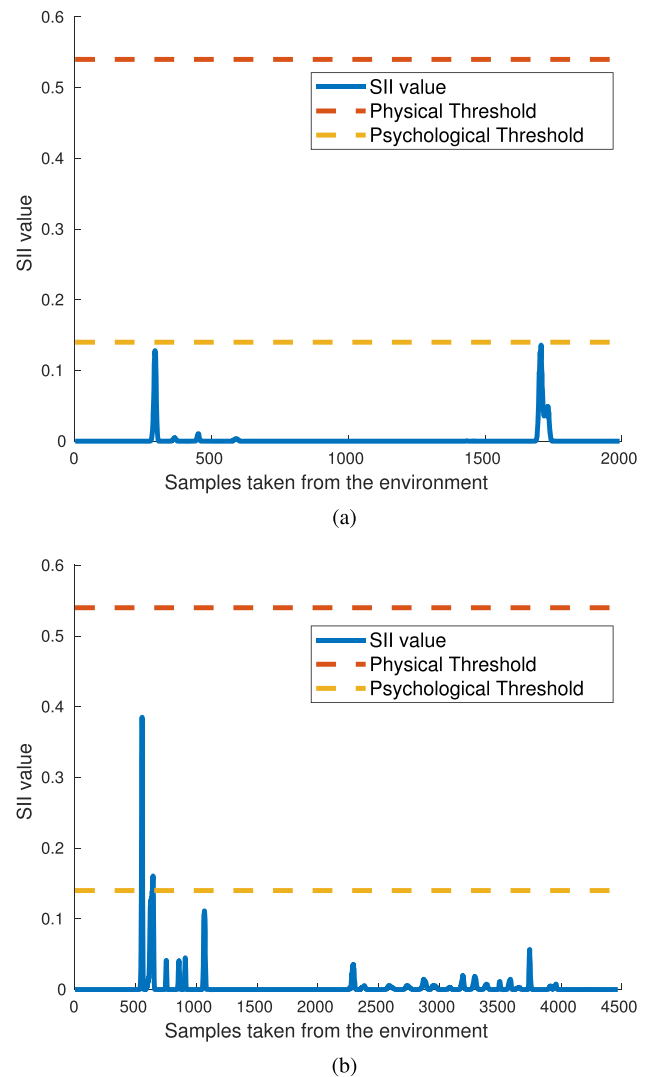
## 5 Results

We examine the robot's motion when the robot is equipped w/out proposed approach for smooth social path tracking and reducing the number of replanning in the dynamic environment. A video clip demonstrating the results of experiments can be found at the hyperlink.<sup>6</sup>

The resulting robot trajectories for each case are presented in Fig. 8b. Table 1 provides a comparison of characteristic path properties of the w/out waypoint approaches.

The results in Table 1 showed that the proposed waypoint approach outperformed the traditional approach in terms of evaluation metrics, which are path execution time, path length, number of replannings, and total heading changes. Total heading changes in angles can be considered as the geometric complexity analysis of the robot path. Therefore, the smoothness of a robot path is quantified using the heading changes. As the number of heading changes decreases, the geometric complexity of the path also decreases, and hence the path becomes smoother. The results indicate that the proposed approach has greatly reduced the number of

<sup>6</sup> <https://youtu.be/gMpVThOJYew>.



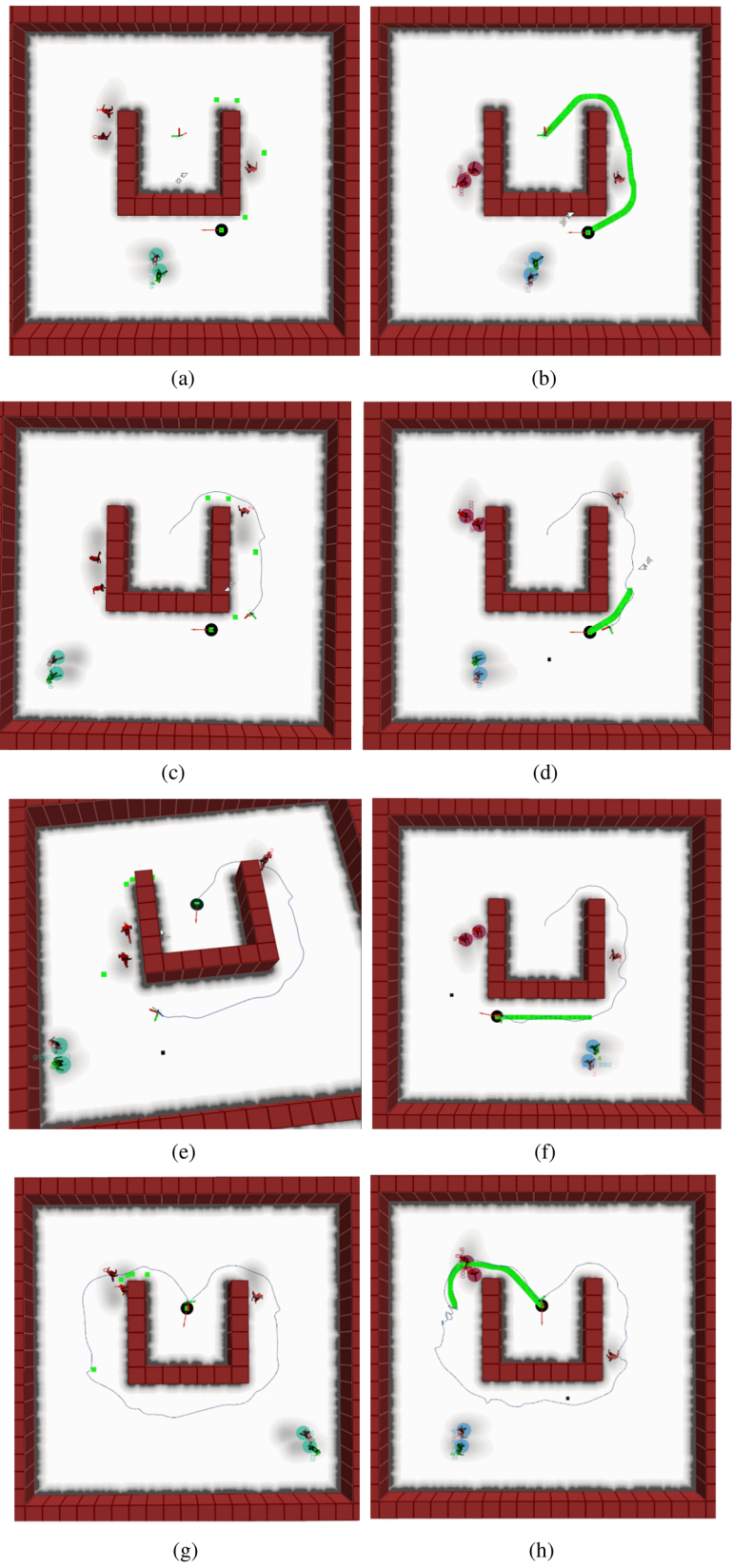
**Fig. 9** SII values of **a** with waypoint selection and **b** without waypoint selection approach experiments

replannings (from 21 to 2 which is 90.4%) and therefore greater performance on the path execution time of the traditional approach by a factor of 53.7% (which reduces from 441 to 204 s). Furthermore, the proposed method generated a 3.62 m (8.3%) shorter path and executed its motion smoother with 2812° which is 55.2% less heading change in degree when compared to without waypoint approach.

In Fig. 9, the SII values are presented. Figure 9a shows that, the SII values are below the determined thresholds for each human the robot encounters, respectively. Therefore, the robot performs social navigation considering social zones, the psychological comfort, and physical safety of humans. On the other hand, due to the absence of waypoint selection in dynamic environments, human comfort is not satisfied for all humans, as seen in Fig. 9b. The sequence of pictures and presentations of the obtained experimental results are



**Fig. 10** Experimental results of navigation illustration. Two cases with **a–c–e–g** and without **b–d–f–h** waypoint selection approach are examined. Green squares represent the waypoints



shown in Fig. 10. Figure 10a–c–e–g illustrates the waypoint selection-based navigation method, while the Fig. 10b–d–f–h demonstrates the approach without waypoint selection. For both cases, it is shown that the robot makes a complete navigation to the goal location and turns back to the starting location, as well as avoiding the obstacles in the case of local minima on the way.

As summary, experimental results show that the behavior of the tradition navigation approach without waypoint selection is characterized by a time costing, inefficient and rough path. In contrast, reducing the unnecessary replanning, improving the human comfort and social acceptance and smoothness of the path are the advantages of the proposed approach. Furthermore, the robot takes shorter paths in less time to reach the goal point.

## 6 Conclusion

Social navigation systems play an important role in human–robot interaction, especially in providing the safety and comfort of the people interacting with robots during navigation in populated environments. SFM would be a good solution to provide the safety and comfortability of people when the robot is not forced to follow a path. However, the local minima problem is a known issue with artificial potential field based algorithms. The problem can be dealt with subdividing the role of motion planning as a two-layered architecture by searching for the global path and executing the plan. In this study, we present a waypoint selection extension to the SFM-based motion planner to ensure a complete (avoiding the local minima), social, and smooth robot control strategy in human–robot interactive social environments. We showed that our method can generate paths that respect people’s social space as well as reducing the unnecessary replanning and provide smoothness during plan execution. Experiments revealed that our approach has advantages over without traditional (without waypoint) planning approach.

Furthermore, while the robot is navigating, it might pass the waypoint over to achieve avoidance of dynamic obstacles in the environment. In that case, the robot either tries to make a new global plan or may switch/change the waypoints according to its current knowledge about the environment. Since this strategy is supposed to occur in the dynamic world, this will be considered as a future work of this study.

**Author contributions** HK and FC took part in the experimentation. SY and HK contributed in the analyses of the results. All authors read and approved the manuscript.

**Funding Information** This research is supported by the Turkish Scientific and Technical Research Council (TUBITAK), partially by Project Nos. 118E214 and 118E215.

**Data availability** Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

## Declaration

**Conflict of interest** The authors declare that they have no conflict of interest.

**Informed consent** Informed consent was obtained from all individual participants included in the study.

## References

- Banisetty, S.B., Forer, S., Yliniemi, L., Nicolescu, M., Feil-Seifer, D.: Socially-aware navigation: a non-linear multi-objective optimization approach. (2019). arXiv preprint [arXiv:1911.04037](https://arxiv.org/abs/1911.04037)
- Ferrer, G., Garrell, A., Sanfeliu, A.: Social-aware robot navigation in urban environments. In: 2013 European Conference on Mobile Robots (ECMR), pp. 331–336. IEEE (2013)
- Ferrer, G., Sanfeliu, A.: Proactive kinodynamic planning using the extended social force model and human motion prediction in urban environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1730–1735. IEEE (2014)
- Ge, S.S., Cui, Y.J.: Dynamic motion planning for mobile robots using potential field method. *Auton. Robots* **13**(3), 207–222 (2002)
- Kirby, R.: Social robot navigation. PhD Thesis, Carnegie Mellon University, Pittsburgh (2010)
- Kivrak, H., Cakmak, F., Kose, H., Yavuz, S.: A multilevel mapping based pedestrian model for social robot navigation tasks in unknown human environments. In: 2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), pp. 1–7. IEEE (2020)
- Kivrak, H., Cakmak, F., Kose, H., Yavuz, S.: Social navigation framework for assistive robots in human inhabited unknown environments. *Eng. Sci. Technol. Int. J.* **24**(2), 284–298 (2020)
- Kivrak, H., Uluer, P., Kose, H., Gumuslu, E., Barkana, D.E., Cakmak, F., Yavuz, S.: Physiological data-based evaluation of a social robot navigation system. In: 2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), pp. 994–999. IEEE (2020)
- Liu, X.H., Zhang, D., Zhang, J., Zhang, T., Zhu, H.: A path planning method based on the particle swarm optimization trained fuzzy neural network algorithm. *Clust. Comput.* **24**(1), 1–15 (2021)
- Lu, D.V.: Contextualized robot navigation. PhD Thesis, Washington University, St. Louis (2014)
- Lu, D.V., Hershberger, D., Smart, W.D.: Layered cost maps for context-sensitive navigation. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), pp. 709–715. IEEE (2014)
- Reddy, A.K., Malviya, V., Kala, R.: Social cues in the autonomous navigation of indoor mobile robots. *Int. J. Soc. Robot.* **13**, 1–24 (2020)
- Regier, P., Shareef, I., Bennewitz, M.: Improving navigation with the social force model by learning a neural network controller in pedestrian crowds. In: European Conference on Mobile Robots (ECMR), pp. 1–6. IEEE (2019)
- Repiso, E., Garrell, A., Sanfeliu, A.: Adaptive side-by-side social robot navigation to approach and interact with people. *Int. J. Soc. Robot.* **12**(4), 909–930 (2020)
- Teso-Fz-Betoño, D., Zulueta, E., Fernandez-Gamiz, U., Saenz-Aguirre, A., Martinez, R.: Predictive dynamic window approach

development with artificial neural fuzzy inference improvement. *Electronics* **8**(9), 935 (2019)

16. Truong, X.T., Ngo, T.D.: Dynamic social zone based mobile robot navigation for human comfortable safety in social environments. *Int. J. Soc. Robot.* **8**(5), 663–684 (2016)
17. Truong, X.T., Ngo, T.D.: Toward socially aware robot navigation in dynamic and crowded environments: a proactive social motion model. *IEEE Trans. Autom. Sci. Eng.* **14**(4), 1743–1760 (2017)
18. Uluer, P., Kose, H., Oz, B.K., Can Aydinalev, T., Barkana, D.E.: Towards an affective robot companion for audiology rehabilitation: how does pepper feel today? In: 2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), pp. 567–572 (2020). <https://doi.org/10.1109/RO-MAN47096.2020.9223534>
19. Yan, Y., Wong, S.: A navigation algorithm of the mobile robot in the indoor and dynamic environment based on the PF-SLAM algorithm. *Clust. Comput.* **22**(6), 14207–14218 (2019)
20. Zanlungo, F., Ikeda, T., Kanda, T.: Social force model with explicit collision prediction. *Europhys. Lett.* **93**(6), 68005 (2011)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Hasan Kivrak** received his M.Sc. and Ph.D. Degrees in Computer Engineering from Istanbul Technical University. He is an Assistant Professor at Karabuk University. He has worked on sign language tutoring robots and affectively aware robot assistants research projects. He is currently working on mobile robots for extreme environments and their digital twin counterparts.



**Furkan Cakmak** is a Ph.D. Candidate in the Computer Engineering Department at Yildiz Technical University. He is currently a Research Assistant and working with the Probabilistic Robotics Group at Yildiz Technical University. His research interests are centered on robot autonomy, particularly path planning, navigation, and exploration algorithms on multi-micro aerial vehicles.



**Hatice Kose** Hatice Kose is a professor at Istanbul Technical University, Turkey, coordinating the Cognitive Social Robotics research group and GAMELab (Game and Interaction Technologies Lab). She is working on Social Assistive Robots, human-robot interaction, affective computing and gamification, mostly in the domain of education, therapy and treatment of children with disabilities. She is part of several national and international projects in this field including FP6 project Robotcub, RISE project Etextweld (Welding of E-Textiles for Interactive Clothing), LUDI COST action COST Action “Play For Children With Disabilities”, and Erasmus+ project EMBOA Affective loop in Socially Assistive Robotics as an intervention tool for children with autism. She conducted several research projects on sign language tutoring robots, affectively aware robot assistants for audiology departments in hospitals, and social robot companions for children with autism.



**Sirma Yavuz** is a Professor of Computer Engineering at Yildiz Technical University. She received her B.E., M.E., and Ph.D. Degrees from Yildiz Technical University. Her research interests include autonomous robots, real-time robot localization, mapping, exploration, and navigation. She is a Founder of the Probabilistic Robotics Group at Yildiz Technical University. In collaboration with her colleagues and students, she has developed teams of real and virtual rescue robots, which have participated in RoboCup championships and received prizes several times. She is a Member of IEEE and ACM.