



Social Elastic Band with Prediction and Anticipation: Enhancing Real-Time Path Trajectory Optimization for Socially Aware Robot Navigation

Gerardo Pérez¹ · Noé Zapata-Cornejo¹ · Pablo Bustos¹ · Pedro Núñez¹

Accepted: 19 March 2024
© The Author(s) 2024

Abstract

As social robots are projected to become an integral part of human life in the coming decades, their ability to adapt movement and trajectory when in proximity to people is essential for ensuring social acceptance during human-robot interaction. A key aspect of this adaptability involves predicting and anticipating human intents during robot navigation. Despite significant strides in the social navigation of autonomous robots within human environments, opportunities for advancements in related algorithms persist. This paper presents a novel real-time path trajectory optimization algorithm for socially aware robot navigation, grounded in the social elastic band concept, incorporating prediction and anticipation of human movements to adapt its forward velocity. Building upon the elastic band framework introduced in the 1990s for adapting robot trajectories in dynamic environments, our proposal of social elastic band differentiates between objects and human presence. This distinction allows for the definition of social interaction spaces and their relationship to the elastic band, facilitating the generation of socially accepted paths that rapidly adapt to environmental changes without causing a disturbance. Integrated into the SNAPE social navigation framework, the algorithm has been tested and validated through simulations and real-world experiments in various environments.

Keywords Social robot navigation · Human-awareness navigation · Elastic-band

1 Introduction

Social robotics has witnessed a surge in investment over the past decade, and their presence in homes and workplaces is expected to become commonplace in the coming years. These social robots are designed with the ability to adapt their behavior as if they were socially aware and integrated,

including the capacity to interact with people, move without disturbing them, and anticipate socially unacceptable situations. Among these capabilities, the ability to navigate environments with humans has garnered significant attention in recent years.

A socially aware robot should navigate the world as a human would, adapting its velocities and trajectories in response to the locations of people around it while avoiding getting too close, interrupting conversations, or making sharp turns close to individuals. The scientific community has devoted considerable attention to the problem of social navigation, with a particular focus on integrating theories like proxemics into robot navigation algorithms. Proxemics theory, introduced by anthropologist Edward T. Hall, describes how humans use and perceive interpersonal distances in their interactions [1]. Researchers employ this theory to create socially aware navigation systems that respect personal space and exhibit socially acceptable behaviors when navigating around people [2, 3]. By considering different spatial zones, such as intimate, personal, social, and public spaces, robots can better understand and adhere to the human perception of

Noé Zapata-Cornejo, Pablo Bustos and Pedro Núñez have contributed equally to this work.

✉ Gerardo Pérez
gperezgonz@unex.es

Noé Zapata-Cornejo
nzapatac@unex.es

Pablo Bustos
pbustos@unex.es

Pedro Núñez
pnuntru@unex.es

¹ RoboLab, Universidad de Extremadura, Avda. Universidad s/n, 10003 Cáceres, Spain

comfort and acceptable distances [3]. Several studies have proposed methods to integrate proxemics into robot navigation, often using cost functions that weigh the importance of maintaining specific interpersonal distances while minimizing other objectives like energy consumption or travel time [4]. In a previous work [2], we introduced a proposal of a social map constructed from the interactions between people and objects and demonstrated how we utilized it for the global trajectory planning of the robot. In this initial approach, people and objects were considered the same type of obstacle during local navigation, which is not always the most appropriate choice, considering that people prefer robots not to pass close by if there is enough space. Machine learning techniques, such as deep learning and reinforcement learning, have also been applied to model human behavior and predict their preferences, further enhancing the robot's ability to maintain appropriate proxemic distances [5].

The challenge in developing a global solution for socially aware robot navigation lies in integrating and real-time collaboration between various subsystems [6]. Each subsystem has its complexities and potential sources of error, making the overall collaboration less straightforward: (i) accurate people detection is crucial, typically involving sensor data fusion from cameras, LiDAR, or other sensors. This process must deal with occlusions, variable lighting conditions, and sensor noise. Additionally, it must be efficient enough to run in real-time to ensure the robot's responsiveness to its environment; (ii) global path planning must consider static and unmapped obstacles while adhering to social conventions such as proxemics. This task may involve incorporating uncertainty in human trajectories and predicting future movements to generate optimal and socially acceptable paths; and (iii) the robot's trajectory adaptation is required to react to sudden environmental changes or unpredictable human behavior. This real-time adaptation involves continuous assessment of the situation, adjusting the trajectory while maintaining safety and social acceptability.

Given the complexities in socially aware navigation outlined above, it becomes imperative to develop frameworks that could integrate many requirements for effective social navigation. In addressing this necessity, we introduced the SNAPE framework in our previous work [7], aiming to provide a comprehensive solution to the challenges encountered in enabling robots to navigate socially enriched environments. The pivotal innovation in [7] concerning the previous one [2] revolved around incorporating human-robot interaction in navigating robots and the high-level strategies of robot actions based on the existing context.

The SNAPE framework, as already hinted at, outlines a comprehensive system for socially aware robot navigation that encompasses, among other aspects, the ability to interact with people, plan socially acceptable paths, and detect movements of people in the environment. Its functionality

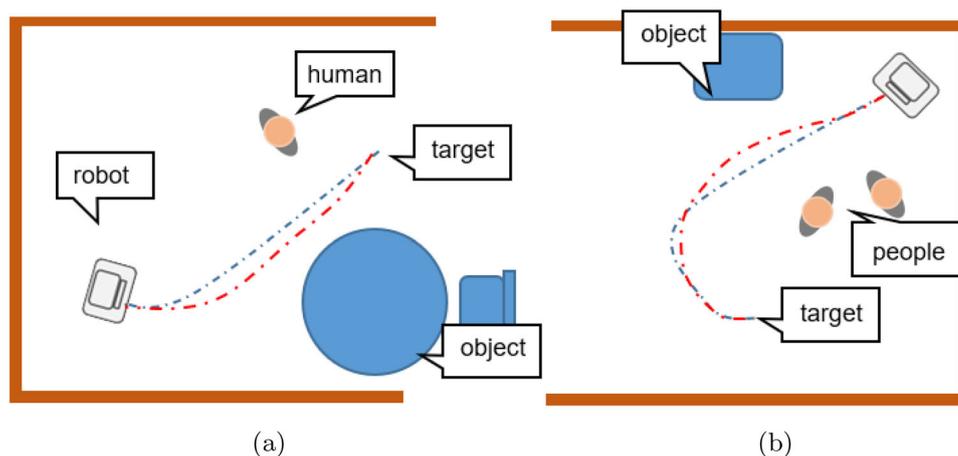
is distributed in five layers, from perception to high-level behavior planning. Although SNAPE discriminates between humans and objects (generating higher cost zones for humans during the global path planning phase), the decisions made when the robot navigates locally may be insufficient from a social acceptance point of view. This is because, in the local planner, the forces exerted by humans and objects on the trajectory are equal for the same distance. In addition, the original SNAPE maintains a velocity independent of the distances to people, which may also lead to a negative perception of the motion. Therefore, the path and motion with the original SNAPE framework can create dangerous situations for people in environments where the robot and people are moving. Figure 1a illustrates an example where a robot navigates in a dynamic environment, with both an object and a person equidistant. In such scenarios, it is reasonable to assume that the robot should prioritize maintaining a greater distance from the person to minimize disruption (red color). A similar situation is depicted in Fig. 1b, where multiple individuals interact simultaneously as the robot navigates its environment. Throughout its trajectory, the robot must adapt its movement based on potential changes in people's positions, moving away from them and attempting to approach objects if that path is viable.

Building on the SNAPE framework, this paper proposes improvements to enhance the autonomous navigation of socially aware robots. The main goal of our work is to refine robot navigation by emphasizing respect for social distances and robot speed control, which are an integral part of social robot behavior. This approach is implemented through adjustments to the social perception and navigation layers of the SNAPE architecture, mainly aimed at predicting the future positions of people and their proximity to the robot's planned trajectory and how this perception influences the robot's speed, and recalibrating the robot's local trajectory parameters by distinguishing between people and objects.

In particular, we present a new social navigation within the SNAPE framework, characterized by three fundamental layers: (i) a new low-level controller that modifies the robot's speed based on the estimated proximity to a person, thus exhibiting a more conservative approach and emulating cautious human-like behavior in the presence of others. We consider it as a basic level of anticipation; (ii) the incorporation of a social elastic band as a component of the local planner, allowing the robot's trajectory to be dynamically adaptive to the movements of people (or objects not accounted for in the global planner), thereby facilitating more responsive and socially harmonious navigation; and (iii) a refined global route planning process that incorporates social norms encapsulated within a social map, constructed based on the robot's environmental perceptions.

The concept of the elastic band, originally conceptualized by [8], serves as a dynamic adaptive layer, allowing the

Fig. 1 Two possible daily scenes: **a** and **b** The path planned by the robot accounts for the distinction between people and objects, with the robot adjusting its trajectory accordingly (red color). In blue color, the path treats people as dynamic objects



robot's globally planned trajectory to mold to the obstacles discerned by proximity sensors. In [2], the foundational algorithm treated all entities in the environment as dynamic, with no distinct consideration for humans. In contrast, the current iteration of our work leverages the SNAPE framework for more nuanced person detection and tracking, enriched social mapping, and optimized path planning, introducing, as a contribution, *a social elastic band that distinctly recognizes and responds to humans and objects, thereby achieving more harmonious navigational outcomes*. This architecture has been assessed in simulated scenarios and validated in real-world experiments, underscoring its practical applicability and effectiveness in socially aware robotic navigation.

This article is structured as follows: Sect. 2 discusses previous works on human-aware robot navigation, highlighting the field's current state and the challenges researchers face. Subsequently, Sect. 3 briefly describes the SNAPE framework, outlining its key components and their respective roles in facilitating socially aware robot navigation. Our paper focuses on the perception and navigation layers, although in this section, all layers are introduced to understand the complete framework. Section 4 introduces our novel contributions, the motion anticipation to people's distance to robots, and the social elastic band concept and explains its implementation and integration with the SNAPE framework. This section delves into the underlying principles of the social elastic band, its ability to distinguish between objects and people, and the resulting improvements in the robot's social behavior. Experimental results are presented in Sect. 5, demonstrating the performance of our proposed algorithm in various simulated and real-world scenarios. We initially include an evaluation of the parameters that affect the social elastic band and a series of tests to subsequently compare the outcomes of our current approach with our previous method, emphasizing its advantages in terms of socially aware navigation and adaptability to environments with people. Finally, Sect. 6 summarizes the main conclusions of this work, high-

lighting the significance of the social elastic band concept in advancing the socially aware robot navigation field.

2 Related Works

Planning a socially acceptable robot's path during navigation is a topic of substantial interest to roboticists, as it plays a crucial role in developing intelligent and socially aware robots. How a robot navigates within natural environments, especially when interacting with humans, dramatically impacts our perception of the robot's intelligence [9]. An explicitly human-aware trajectory should consider a combination of factors, such as minimizing distance traveled, reducing energy consumption, and adhering to social constraints like maintaining a comfortable distance from people or avoiding disturbing ongoing conversations. These insights and conclusions stem from a comprehensive analysis of the current state of the art, including surveys by [3, 4, 10], and more recent contributions, such as the one presented by [6].

Human-aware navigation frameworks encounter a considerable challenge when attempting to combine deliberative and reactive behaviors during navigation. A social robot must be able to react effectively to the dynamic environment, including moving people, while simultaneously planning a socially accepted path [11]. Path planning, a classic problem in autonomous robot navigation, has been extensively studied in the literature [12, 13]. The objective is to determine a set of waypoints the robot approaches sequentially, optimizing its performance based on a global objective function or cost function, such as the shortest or quickest collision-free path. Several methods have been proposed in the literature to address this problem in static scenarios. One such approach is integrating human spatial models, such as proxemics, into the robot's global planning process to determine socially appropriate paths [14]. In [15], Sisbot et al. introduced a path-planning algorithm that considers humans'

comfort zones, adapting the robot's trajectory to maintain a comfortable distance from people in the environment. Similarly, [10] presented a survey on socially aware robot navigation, emphasizing the importance of respecting human social norms during path planning. Another critical factor to consider is the predictability of robot motion. In [16], the authors proposed a path-planning algorithm that generates predictable and legible robot trajectories, making it easier for humans to understand the robot's intentions. Moreover, [17] introduced a socially adaptive path-planning approach that learns from human demonstrations, adapting the robot's motion to follow social conventions.

In dynamic contexts, robots adapt their motion plans to address both global and local path planning, with human-aware navigation complicating this classic problem by integrating environments filled with people, thus necessitating advanced approaches for developing both path types. The inherent challenges of navigating around humans, such as respecting social norms and modifying trajectories based on human movements, are mainly addressed by the local path, given its role in managing dynamically changing elements, while the global path provides a stable foundational route based on relatively constant maps [10]. This hierarchical approach enables the navigation system to maintain efficiency and adaptability by minimizing computational load and adjusting to spontaneous environmental changes, facilitating robust navigation in human-populated environments even without the precise anticipation of individuals' future movements.

Numerous widely-used solutions for these short-term plans have been developed in the field of robotics, some of which include the classic Dynamic Window Approach (DWA) [18], Potential field [19], Elastic bands [8], Reciprocal Velocity Obstacles (RVO) [20], and Social forces models (SFM) [21]. These approaches have been essential in addressing various aspects of local path planning in dynamic environments. In a recent study by [22], the authors offer a comprehensive overview of the latest path-planning strategies, clearly distinguishing between classical hierarchical planners, which involve global and local path-planners, and reinforcement learning-based approaches that leverage advanced learning techniques. From the analysis of these works, the authors conclude that no single optimal or sub-optimal strategy applies to all scenarios, highlighting that the field of path planning still has ample room for improvement and the development of innovative methods that can effectively tackle the challenges of dynamic environments [22].

In recent years, the field of socially aware robot navigation has seen remarkable developments, evolving predominantly into two main paradigms: machine learning-based and model-based approaches. Machine learning-based approaches are used in anticipating human behaviors [23], [24]. These strategies employ machine learning techniques,

like deep reinforcement learning [25], to predict real-time human intentions, which helps modify the robot's trajectory. For instance, Kobayashi et al. [26] have exploited deep learning to navigate robots in high-density human environments and to predict human interaction to prevent collisions and ensure harmonious navigation. Peltzer et al. [27] emphasized integrating human path preferences in robot navigation amidst obstacles using a stochastic observation model for humans, ensuring minimal interventions. Although machine learning-based approaches, exemplified by previous works, have shown advances in predicting human intentions and robot navigation in densely populated environments, they carry significant disadvantages compared to model-based approaches. Their heavy reliance on large datasets, high computational demands, lack of interpretability, susceptibility to input variations, and difficulties in generalizing to new scenarios highlight the inherent limitations of these strategies. The model-based approach proposed in our paper addresses these issues by leveraging theoretical foundations to provide a more computationally efficient, interpretable, robust, and generalizable solution.

Despite these advancements in machine learning-based approaches, model-based approaches, showcased in works like those by Kitagawa et al. [28], Singamaneni et al. [29], and Kollmitz et al. [30], are equally prominent and advanced, offering nuanced solutions for socially aware navigation. For example, Kitagawa et al. developed human-inspired motion planning for omnidirectional social robots to achieve more natural human interactions. Singamaneni et al. addressed the challenge of invisible humans in social navigation, providing innovative insights into dealing with unforeseen human behaviors. At the same time, Kollmitz et al. explored learning human-aware robot navigation from physical interactions via inverse reinforcement learning. These works represent recent strides in model-based approaches to socially aware navigation. Our paper introduces a novel contribution through a distinctive methodology, emphasizing a social elastic band that modifies the trajectory by conforming to interaction spaces. It anticipates potential scenarios wherein the robot approaches an individual, utilizing a basic technique manifested as a reduction in speed. In the work presented in [31], the authors introduce an adaptive, socially aware navigation system that enables a robot to approach groups of humans in a socially acceptable manner, dynamically adjusting personal and group space parameters based on the group arrangement and space constraints. Unlike our work, [31] focuses solely on the more reactive navigation. Furthermore, other approaches have also significantly contributed to model-based approaches in social aware navigation, with works such as Ferrer and Sanfeliu [32] and Repiso et al. [33] offering anticipative solutions and adaptive planners for dynamic real-life environments.

Specifically, the authors in [32], engage in anticipative kinodynamic planning to enable effective navigation in urban and dynamic environments. The more advanced work in [33] refines the Extended Social Force Model (ESFM) to facilitate closer interaction with people, enhancing the adaptability of the planner in diverse socially aware navigation contexts such as solo navigation, accompaniment of one or several people, approaching, and simultaneous accompaniment and approaching. These studies are very well-rounded, particularly the work presented in [33], which employs what they term an Adaptive Social Planner—a fusion of Rapid Randomly Tree (RRT) and ESFM for various formations. Our contribution resembles the utilization of an innovative navigation framework featuring a global planner that seeks the optimal route (a capability not intrinsic to RRT) and a local planner. Instead of modifying the trajectory according to a social force model, we rely on the Elastic Band method, a technique proven to efficiently smooth paths and adeptly manage dynamic elements, those not represented on the map. Our paper is not predicated on accompaniment but on navigation within an environment inhabited by humans who should not be disturbed during the navigation.

Each of the aforementioned studies has proposed a solution to socially conscious navigation, a challenge also addressed by the research presented in this paper. This work introduces an innovative strategy, seamlessly integrated into the SNAPE framework, that enhances robotic navigation performance within human-inhabited environments. It achieves this by discerning between moving objects and humans, refining how such differentiation impacts the local planner, and modulating robot speeds in real time according to the prevailing context. Moreover, our approach incorporates a model-based methodology, resonating with concurrent studies in this domain. The synergy with the SNAPE framework propels progress toward a solution where robots can proficiently navigate and engage with humans to accomplish their objectives.

3 SNAPE: Social Navigation Framework Over CORTEX

Figure 2 illustrates the original SNAPE navigation framework schematic, outlined in [7], and the main changes developed in this paper to optimize human-aware robot navigation. The SNAPE framework was structured into five discernible layers: (1) perception layer; (2) social layer; (3) navigation layer; (4) HRI interaction layer; and (5) planning layer. These layers are depicted in the original schematic, to the left of Fig. 2. We display the principal modifications introduced in this article within the same figure to the right. Firstly, we alter the perception layer to predict people's positions in the robot's proximity and subsequently modify the navigation

layer. Within this layer, prior estimates of people's positions are incorporated to adjust the robot's velocities based on the distance of these estimates to the robot's path within the time horizon. We also integrate a social elastic band that differentiates objects and humans in the local planner. A more detailed description of Fig. 2 is given in the following subsections.

The CORTEX cognitive architecture underlines the construction of SNAPE, as proposed in [34]. Figure 3 illustrates a conceptual schematic of CORTEX, highlighting the relationship between its software agents and the layers of SNAPE. CORTEX is assembled with independent software agents, each maintaining a shared Working Memory (WM).¹ CORTEX defines a software agent as an entity capable of interacting with the WM, synchronizing the outcomes of its actions with other agents through this WM. This WM encompasses all the geometric (e.g., positions of the robot and people) and symbolic information (e.g., interacting, in, or connected) representing the robot's knowledge. The five independent layers of SNAPE are each orchestrated by one or multiple software agents within CORTEX, exchanging information concerning the robot and its environment. For instance, the perception layer comprises several independent agents tasked with responsibilities such as people detection, tracking and object detection.

In formal terms, the WM is a directed graph represented as a pair $G = (V, E, \omega)$, consisting of V , a collection of vertices; E , a collection of edges; and an incidence function that maps each edge to an ordered pair of vertices, $\omega : E \mapsto (x, y) | (x, y) \in V^2 \wedge x \neq y$. encompass instances of concepts recognized by the system. As an illustrative example, the right section of 3 shows a use case involving a robot in a nursing home. The representation of this world within CORTEX is shown as a *Working Memory* in the left part of the Fig. 3. These concepts can relate to either physical or internal entities. When an agent instantiates a physical concept into the WM, it must maintain the anchor to its world entity throughout its existence. Agents generate internal entities, such as missions, intentions, or plans, and represent them in the graph for other agents to recognize. This graph is a live illustration of the context relevant to the ongoing mission. Relating the aforementioned terms to Fig. 3, V is constituted by the nodes that physically represent the world: people, rooms, objects, and E encompasses the relations between nodes, which can be either geometric (e.g., Rotation-translation links) or symbolic (e.g., in, connected, or interacting).

Having illustrated the overarching structure and foundational components of the SNAPE framework and highlighted the integrative role of CORTEX's Working Memory in

¹ The WM, also referred to as Deep State Representation (DSR), is capable of accommodating diverse data at multiple levels of abstraction [35].

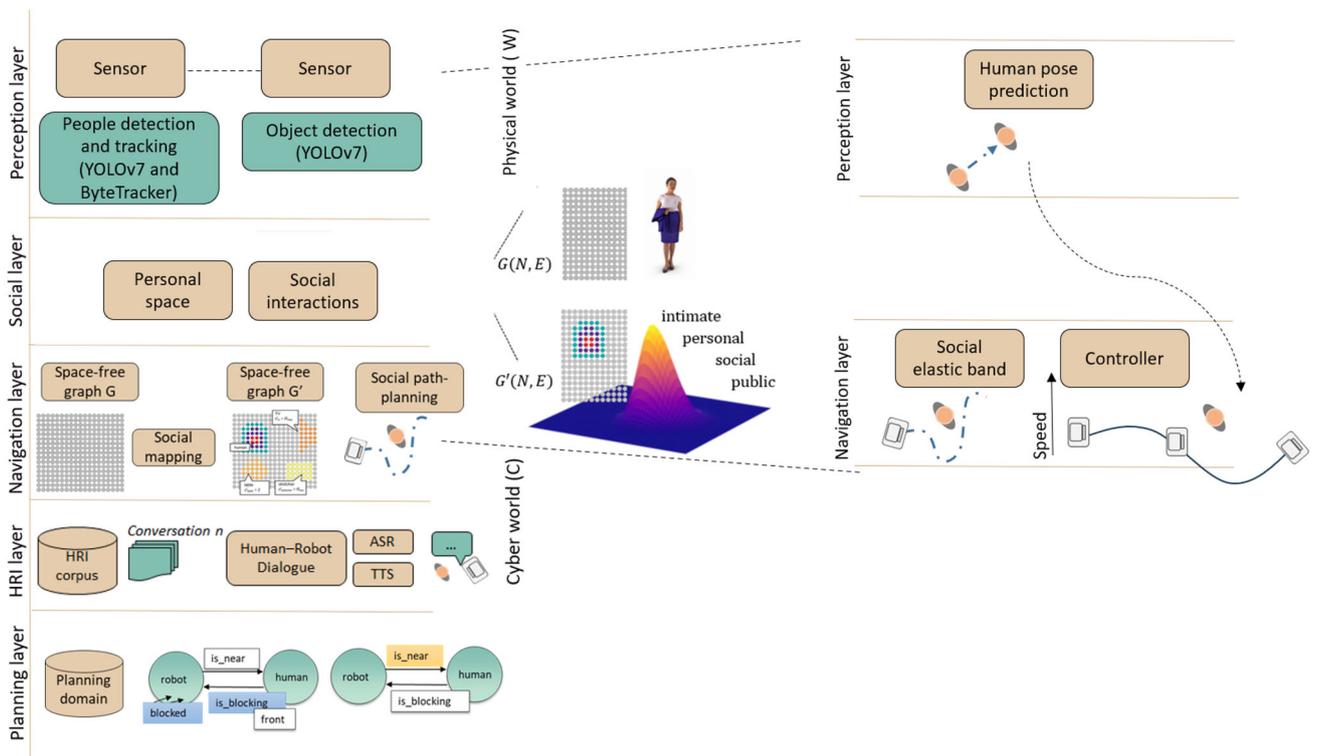


Fig. 2 Overview of the SNAPE framework [7]

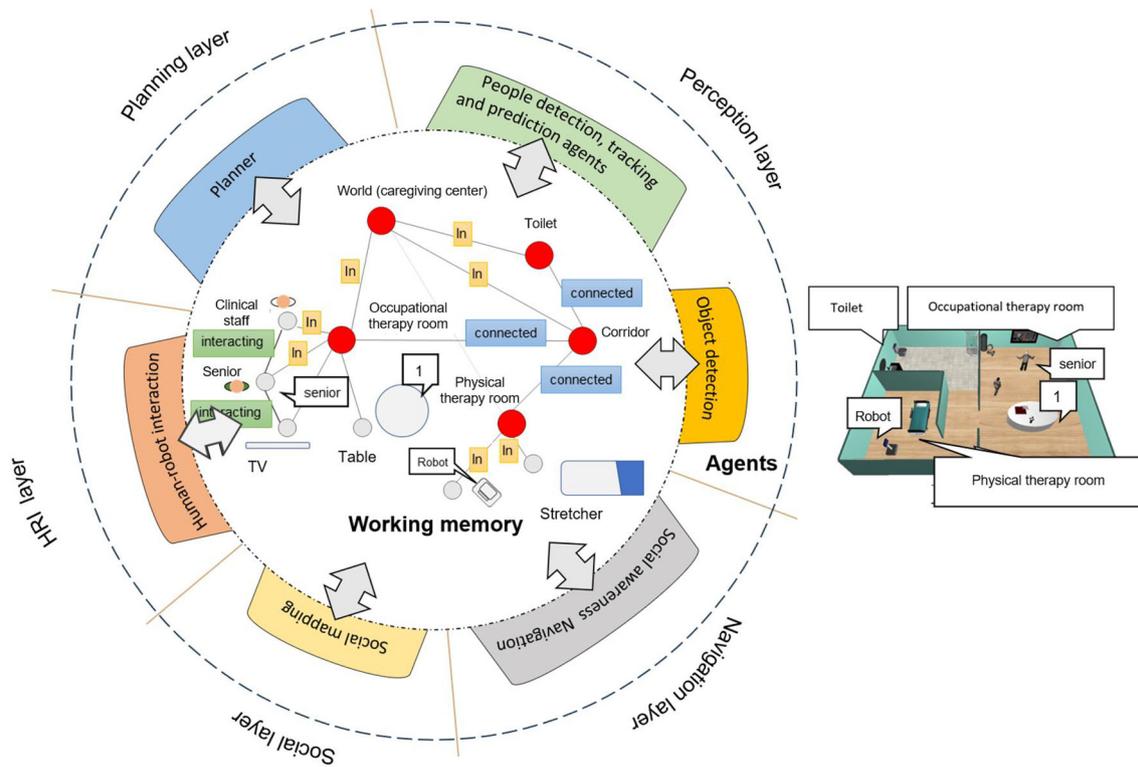


Fig. 3 A schematic view of the CORTEX architecture

enabling effective multi-agent interaction, we will now proceed to delve into a description of each distinct layer within SNAPE. This article mainly focuses on the perception and navigation layers and their vital role in achieving effective human-aware robot navigation, including basic anticipating human movements.

3.1 Perception Layer

The perception layer is an essential tool in socially aware robot navigation systems, furnishing the navigation layer with information about the robot's surroundings to inform socially acceptable decision-making. This layer detects various environmental elements, such as the position and pose of people or objects, ensuring the robot can navigate without collisions. For our current approach it accurately discerns and tracks people's poses in real-time, a task that is particularly challenging in dynamic environments, thereby aiding the navigation core in anticipating the movements and behaviors of people around it.

In the proposed approach, the perception layer consists of an integrated sensor system that enables the robot to perceive its environment. It could be composed of various sensors, including cameras, microphones, or tactile devices, that allow the robot to gather information about its surroundings (Fig. 2). These sensors can be internal to the robot or external, configuring a physical world to detect people and objects. A specific software agent tracks the position and orientation of each person, $h_i = (x, y, \theta)_i$, while the pose of each object, o_j , is described by $p_{o_j} = (x, y, \theta)_j$. The perception layer detects objects and people in the environment, which are assumed to be detected by the agents in CORTEX. The sets $H_n = \{h_1, h_2 \dots h_n\}$ and $O_m = \{o_1, o_2 \dots o_m\}$ represent the detected humans and objects, respectively. To ensure that all agents in the architecture share the same knowledge about the robot's surroundings, this information is updated in the WM [2, 11].

To detect and track individuals, the agent employs a method that takes the output from the YOLOv7 Deep Neural Network (DNN) [36] and feeds it into the ByteTracker [37] tracking algorithm. The system takes, as input at time τ , a color image T with dimensions $u \times v$ and generates, as output, the 3D location and orientation of each person in the image h_i , along with their track identifier id_i . The YOLOv7 processes the image and returns the Regions of Interest (ROIs) of people $B_i = \{a_i, b_i\}$, where a_i and b_i represent the top-left and bottom-right points, respectively. Upon detecting a person, a track is initiated using the ByteTracker algorithm, acquiring the id_i . The 3D pose from the robot's point of view is obtained by analyzing the central pixel value of the ROI within the stereo image. To clarify, h_i represents the centroid of the person detected, calculated based on the bounding box

coordinates, a_i and b_i (Eq. 1).

$$h_i = \left\{ a_{x_i} + \frac{b_{x_i} - a_{x_i}}{2}, a_{y_i} + \frac{b_{y_i} - a_{y_i}}{2} \right\} \quad (1)$$

Rather than solely relying on the vector H_n , our algorithm estimates the future positions of people, H_n^e (where the superindex e denotes estimated), to better anticipate dynamic environments. To accomplish this, we first calculate the velocity vector for each person near the robot. Let h_i^τ , and $h_i^{\tau+\Delta\tau}$, represent the positions of human i at two distinct moments in time. The velocity vector is then defined as:

$$\vec{v}_{h_i} = \frac{h_i^{\tau+\Delta\tau} - h_i^\tau}{\Delta\tau} \quad (2)$$

This velocity vector is characterized by both its magnitude $|\vec{v}_{h_i}|$ and direction $\beta_{v_{h_i}}$. Utilizing this information, we can determine the estimated positions of each person, h_i^e , at a specific time interval t . Instead of relying solely on two discrete poses, our algorithm employs a temporal window of prior poses $h_i^{(\tau-k)}$, $h_i^{(\tau-(n-1))}$, ..., $h_i^{(\tau-1)}$, h_i^τ to estimate the velocity vector of each person near the robot with greater accuracy. This consideration of multiple poses within a time window allows for a nuanced understanding of each person's movements, accommodating sudden and unpredictable changes in direction or velocity, which is essential for real-world applications where human movements are inherently dynamic. It is important to clarify that this estimation will only be used to anticipate the presence of moving people near the robot during navigation, modifying its speed to adapt to the context socially, where values of $t = 2s$ shall be used.

3.2 Social Layer

The second layer of the proposed architecture focuses on the robot's social awareness, which is grounded in generally accepted social norms. Social robots must adhere to social norms, such as maintaining suitable social distances, refraining from interrupting conversations, and seeking permission before intervening during navigation. At the heart of the social awareness layer is the notion of a social interaction space, delineated by individuals in the environment based on their personal space and proxemics theories, extending to interaction space with objects [2]. The following is an overview of these concepts and their use in the context of this article.

After determining the positions of humans in the environment, H_n , and recognizing the static objects in the environment O_m , this layer constitutes a social representation of the space. Figure 2 illustrates the elements that make up this layer: on one hand, we initiate with information regarding people and objects in the environment. This information,

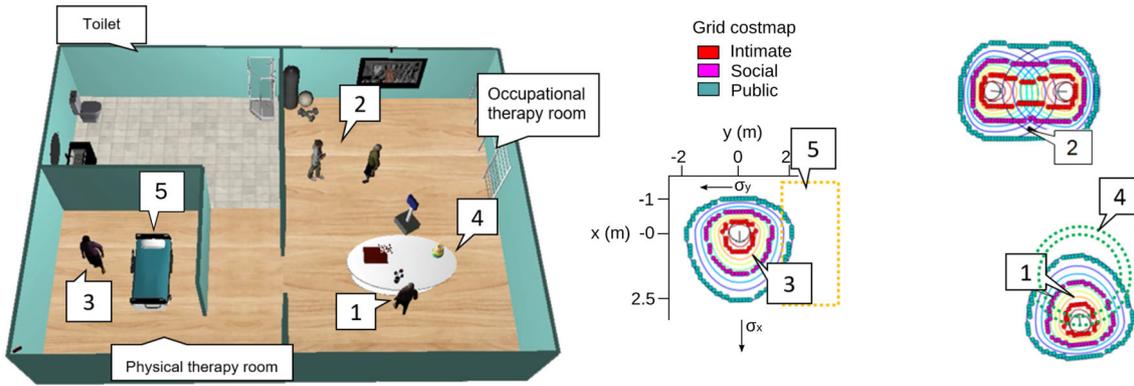


Fig. 4 The figure shows a nursing environment, including the positions of people and objects. On the right, labelled, are the interaction spaces of each element, using values of $\sigma_x = 2$ and $\sigma_y = 4/3$. The costs

entered in the grid are equivalent to the values 8, 10 and 100 for public, social and intimate spaces respectively

via semantic models and social norms, alters the robot's original map. A social map replicates the robot's free space map, enriched with social information. Herein, we integrate the concepts of accessibility and weight associated with each grid in the original map [2]. The accessibility of a node is a Boolean variable, with a value of 1 if the space is unoccupied and 0 otherwise. The weight, w_i , represents the traversal cost of a node, illustrating the effort required for the robot to reach that node (higher values of w_i imply that the robot should circumvent this route).

In our model, proxemics theory defines zones (intimate, personal, social, and public) that modify the original grid map by assigning varying weights and accessibility to each zone [2]. Initially, a free space map is created to pinpoint static impediments. This grid map is then modified to incorporate social interaction spaces. For an individual, this space is delineated by contour lines of an asymmetric Gaussian. The model is grounded in a summation of Gaussians in scenarios involving interactions between multiple individuals. When individuals interact with objects, this space is represented by specific contour lines depending on the object's shape. Figure 4 depicts various interaction spaces and how they impact the original grid's weight. To the left, Fig. 4 displays a nursing environment where the robot will operate. This environment contains information about the static objects in the rooms (*i.e.*, a stretcher and a table for activities). The perception layer identifies people in the environment. In the social layer, interaction spaces are introduced for individuals and objects with which individuals may interact. These spaces directly influence the weight and accessibility of the grid.

Formally, the robot's environment is modeled by the grid $\Gamma(\gamma, \epsilon)$ of γ cells and ϵ links, evenly distributed in the space. Each cell γ_i has two different parameters: accessibility, a_n , and weight, w_n . In the beginning, all boxes have the same weight, 1. Later, Γ is actualized to a new grid $\Gamma'(\gamma, \epsilon)$ where we add all the social interaction spaces around people [2].

The region surrounding each individual is modeled as an asymmetric Gaussian function. The layer also identifies and maps interactions between individuals, such as conversations, using a spatial density function that the robot should not traverse. Finally, the framework characterizes the interaction space between individuals and objects. The personal space of each h_i is modeled using an asymmetric 2-D Gaussian curve $g_{h_i}(x, y)$ as defined in Eq. 3.

$$g_{h_i}(x, y) = e^{-\left(\frac{(x-x_i)^2}{2\sigma_{x_i}^2} + \frac{(y-y_i)^2}{2\sigma_{y_i}^2} - \frac{2k_{xy}(x-x_i)(y-y_i)}{2\sigma_{x_i}\sigma_{y_i}}\right)} \quad (3)$$

where σ_{x_i} and σ_{y_i} are the standard deviations along the x and y axes, respectively, and k_{xy} is a coefficient that controls the correlation between the two axes. These values depend on the orientation θ_i of the person and are used to emphasize the region in front of the person, as defined by the theory of proxemics [2].

The algorithm then clusters people in the environment based on their distances using a Gaussian Mixture [2]. The personal space functions of each individual are summed to obtain a global interaction space $G(h)$, as shown in Eq. 4.

$$G(h) = \sum_{i=1}^n g_{h_i}(x, y) = \sum_{i=1}^n e^{-\left(\frac{(x-x_i)^2}{2\sigma_{x_i}^2} + \frac{(y-y_i)^2}{2\sigma_{y_i}^2} - \frac{2k_{xy}(x-x_i)(y-y_i)}{2\sigma_{x_i}\sigma_{y_i}}\right)}, \quad (4)$$

being n is the number of people. To incorporate the space affordances of objects in the environment, we store the interaction space i_{o_k} of each object $o_j \in O_M$ as an attribute. The interaction space i_{o_k} captures the space required to interact with the object and varies for different objects in the environment [2]. For example, in objects such as posters or screens, the interaction space follows the format of a trapezoid, while

other objects such as tables and beds maintain the same format as the original object, circular or rectangular.

The aggregation of regions outlines zones in which navigation should be curtailed or limited, and the final step involves updating the free space graph Γ values. These restricted areas' boundaries consist of k polygonal chains, or polylines, represented as $L_k = l_1, \dots, l_k$, where k indicates the total number of interaction regions. Each curve, l_i , is formulated using the expression $l_i = a_1, \dots, a_m$, with the vertices $a_i = (x, y)_i$ of the curve positioned along the region's periphery. Our method incorporates distances determined by proxemics-intimate, personal, social, and public-established relative to a person's center.

Within the area formed by $L_k^{intimate}$, the accessibility a_i of all nodes $N_i \in \Gamma$ is designated as occupied, or $a_i = occupied$. This restriction effectively prevents the robot from invading this space, averting disturbance to the individual. For personal and social spaces, the graph nodes' accessibility remains unchanged; however, the associated weights will be modified. The weights w_i of all nodes $\gamma_i \in \Gamma$, enclosed by the spaces formed by $L_k^{personal}$ and L_k^{social} , are adjusted, resulting in higher costs in the personal area compared to the social area. Lastly, the remaining graph represents the public space with unaltered weights.

To illustrate our social mapping definition, we present two figures. In Fig. 4, a nursing environment is depicted, showcasing the positions of individuals and objects. To the right, interaction spaces for each element are labeled. Different colors within the figure represent varying weights in the grid. Besides, a 3D perspective of the simulated scenarios is pre-

sented in Fig. 5a. Figure 5b depicts the outcome of applying our social mapping technique to these scenarios, with social interaction spaces represented in various colors. In Fig. 5b, it is evident that individuals h_1 and h_2 engage in interaction at the bottom, as their personal spaces overlap. Furthermore, objects within the environment are incorporated into social mapping by including their respective interaction spaces.

3.3 Navigation Layer

The subsequent stage involves planning a socially acceptable path and navigating towards the target. This step is characterized as a three-level hierarchy within the framework. As depicted in Fig. 6, these levels encompass:

- Path planning algorithm: The path planning algorithm utilizes the social map Γ' to generate global trajectories towards specific targets. This algorithm can be linked to the Dijkstra algorithm's methodology. It creates a graph in which each point in the environment is a node, and the edges are links between nodes. The distance between the two points determines the weight w_i of each node γ_i . At this juncture, the global planner takes the information from the social map and computes a trajectory for the robot between two distant points, potentially separated by several rooms. If the robot knows the positions of people and objects, this trajectory can choose those spaces to navigate where the robot is less disruptive. We want to emphasize that incorporating human interactions

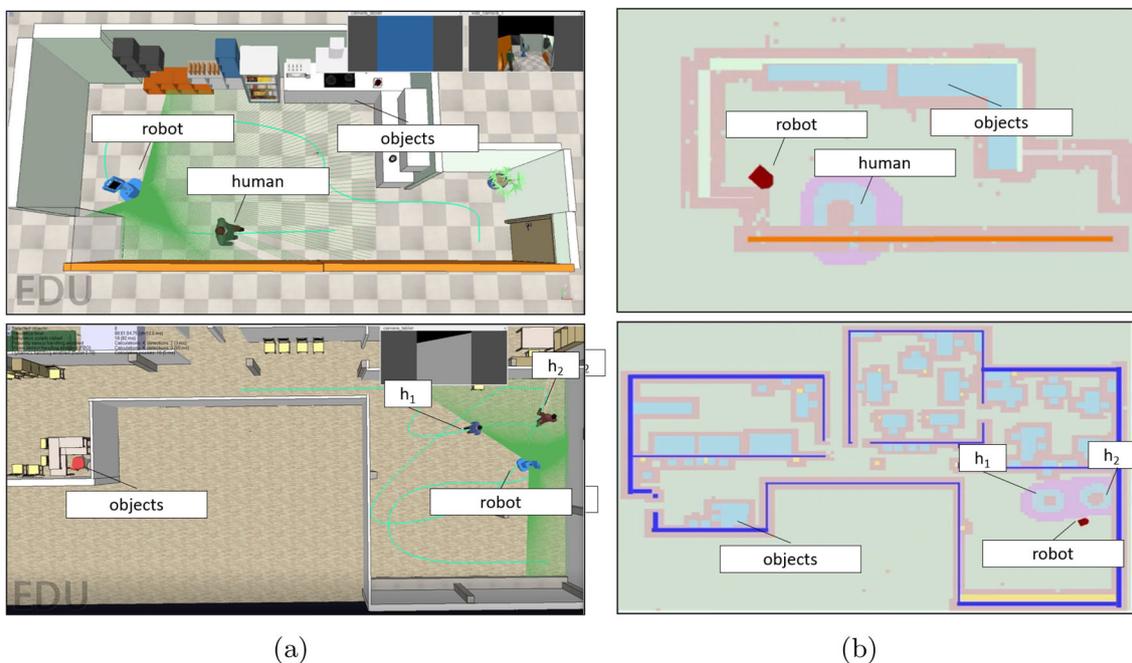


Fig. 5 a Simulated environment with people and object; b social mapping of the environment, showing social interaction spaces

and their locations, especially utilizing external sensors, in the global planning phase is fundamental to enabling the robot to proactively adapt its path, ensuring socially-aware navigation from the onset and mitigating potential disruptions in diverse and dynamic environments.

- **Social elastic bands:** To accommodate local changes detected by range sensors (*i.e.*, people moving around the robot), the planned path is deformed in real-time. At this point, the algorithm modifies the points of the original path using the information derived from the curves $L_k^{intimate}$, $L_k^{personal}$, and L_k^{social} of the individuals who are near the robot's trajectory. A deformable, collision-free path called an elastic band algorithm was first introduced in Quinlan's work in 1993 [8]. The algorithm is based on imaginary forces acting on the points along the path. These forces are divided into the internal contraction force (f_c) and the external repulsion force (f_r). The contraction force removes slack from the path, while the repulsion force guides the robot around obstacles. In this work, we present a novel repulsion force (f_s) that considers the social interaction spaces of people around the robot, predicting the position and anticipating the robot's movements. To understand the concept of social elastic bands, imagine a robot moving along a path with three successive points (p_{i-1} , p_i , and p_{i+1}). These points determine the contraction force, while the range sensors provide readings that determine the repulsive force. When the framework predicts that a person k will enter the robot's path, their interaction spaces ($L_k^{intimate}$, $L_k^{personal}$, and L_k^{social}) dictate the social force acting on the elastic band. For this purpose, the sensor reading (*i.e.*, laser) is modified at those points corresponding with points of the polygons $L_i(x, y)$. These three forces, f_c , f_r , and f_s , work together to influence the final position of the path point p_i^{t+1} (see Fig. 7). The social elastic band algorithm proposed in this work extends the original elastic band algorithm to handle better complex real-world environments that involve human-robot interactions. This algorithm is described in the next section.
- **Control:** The elastic band algorithm smooths the robot's path and serves as a basis for the feedback control law that guides the robot. The control law adjusts the robot's velocity based on its anticipation behavior. In other words, the robot's velocity is modified according to the distance to individuals in the future. The robot initially aligns itself with the next point on the trajectory and then adjusts its forward speed. This forward velocity v_f is multiplied by a gain κ^{v_f} defined by a sigmoid between 0 and 1:

$$\kappa^{v_f} = 2 / \left(1 + e^{\frac{(d_h^t - d_{p_i^{t+\Delta\tau}})^{\lambda}}{d_{p_i^{t+\Delta\tau}}^{\lambda}}} \right) - 1 \quad (5)$$

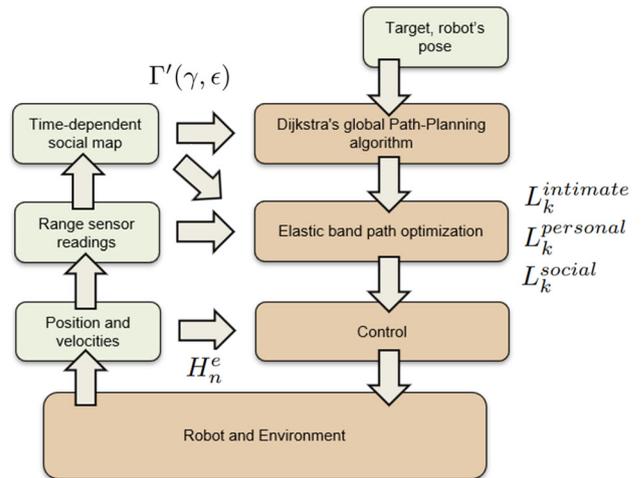


Fig. 6 The navigation module features a tri-level hierarchy, delineating the arrangement and coordination of the distinct elements that contribute to the robot's navigation procedures

where d_h^t is the distance to the nearest individual h_i^e , $d_{p_i^{t+\Delta\tau}}$ is the distance to the point in the robot's trajectory at instant time $t + \Delta\tau$, and the gain λ is associated with the slope of the sigmoid. The parameter λ influences the slope of the sigmoid; a higher value of λ makes the transition of the sigmoid function more abrupt, whereas a lower value makes it smoother. For the purpose of our experiments and the context of our model, λ is assigned a fixed value of 0.001, ensuring a balanced transition in the sigmoid function. This equation controls the robot's velocity along the elastic band, ensuring it follows the planned path while avoiding collisions with obstacles and anticipating human movements. Therefore, the control law complements the social elastic band algorithm, making it possible to handle social aspects in path planning for mobile robots.

3.4 Human–Robot Interaction Layer

The development of socially-aware robots capable of navigating complex human-populated environments necessitates the ability to engage in meaningful interactions with humans. The SNAPE framework addresses this requirement by introducing the interaction layer. This layer is designed to facilitate specific interactions (*e.g.*, Automatic speech recognition, Text to Speech algorithm, dialogue management, among others) that may arise during the navigation process, enhancing the robot's social awareness and fostering a more intuitive human–robot collaboration [7]. However, it's important to clarify that this paper does not delve into a detailed exposition of this interaction layer as it falls outside the scope of our current work.

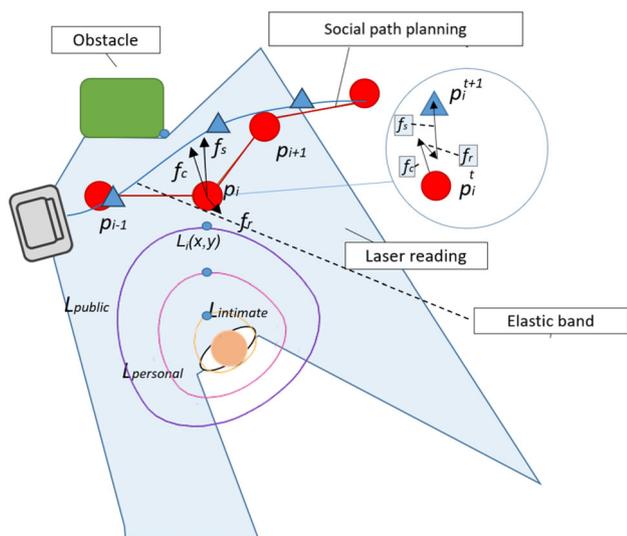


Fig. 7 The basic elastic band algorithm utilizes imaginary forces of contraction and repulsion, represented by f_c and f_r , respectively, to adjust the band until it reaches equilibrium. In this work, we enhance the original algorithm by incorporating the social force f_s that considers the individual’s personal interaction spaces. The magnitude of each force is calculated based on the distance between the points along the robot’s trajectory. In the figure, we can observe the global path in red and the trajectory ultimately followed by the robot in blue

3.5 Planning Layer

The final layer of our proposed architecture addresses the robot’s need to plan specific actions to navigate toward its target successfully. The action planning process within the context of human-robot interaction for navigation tasks involves identifying key elements in the planning problem: an initial world model, a mission, and a set of actions (*i.e.*, the planning domain). In the SNAPE framework, the planning process relies on the symbolic information contained within the WM, which utilizes representation nodes as symbols and graph edges as predicates [7]. To illustrate, a symbol in this context might represent a specific entity or state in the environment, such as a person, an object, or a location. On the other hand, a predicate could represent relationships or properties between these symbols, perhaps denoting the distance between two objects or whether a particular state is achievable from the current configuration. For example, if a node symbolizes a location ‘room’, and another node symbolizes our robot ‘robot’, a predicate connecting these nodes could represent the relation “robot is in the room”, illustrating the presence of the ‘robot’ in the location ‘room’. Figure 8 represents the original and final states after performing the actions. In the final state, the robot must change rooms. To accomplish the task of changing rooms, the robot is required to execute a sequence of actions, each one meticulously defined within the planning domain, where this planning domain serves as a repository of feasible actions available to the robot, structured

around the system’s rules and the environment’s constraints. Similarly to the preceding layer, it is necessary to make clear that this layer does not contain any novel contributions in the current article.

4 Path Optimization Algorithm for Human-Aware Robot Navigation

This section proposes a navigation strategy wherein the robot’s trajectory is represented as a sequence of 2D points, denoted as bubbles (x, y) , forming an elastic band. This elastic band is subjected to artificial forces, which deforms in real time to a short and smooth path that maintains clearance from the obstacles [8]. The trajectory is defined as an ordered collection of points, $P = p_i : p \in \mathbb{R}^2 \times \mathbb{N}, i \in 0..N$, encompassing two real coordinates representing the robot’s global position and an integer signifying the bubble’s radius. The radius is determined by the minimum distance to surrounding obstacles, as detected by sensors such as lidar, sonar, or vision-based systems and by the distance to individuals in the robot’s path. The radius calculation function, $\rho(p)$, is defined as $\mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \{\mathbb{R}^+ \cup 0\}$ and is implemented through exploration over the laser array and the set of visible points, and also considering the distance to people. Further elaboration on the forces and processes impacting the path is provided below.

- The path planning algorithm requires the robot’s location, the goal position, and a time-dependent social map. As outlined earlier, our approach to global path planning leverages the established Dijkstra algorithm, which generates the robot’s path, focusing on dealing with static obstacles and interaction spaces in the social map.
- Upon establishing an initial path, the number of points in P undergoes continuous modification through the addition and removal of elements. The objective is to preserve a uniform distribution of points along the band, irrespective of expansion or contraction. A new point is incorporated if the distance between two consecutive points surpasses half the robot’s length. In contrast, two points are amalgamated into one if their separation is less than half the robot’s length. This dynamic adjustment guarantees a smooth path for the robot, adapting to environmental changes as needed. Let $d_{i,i+1}$ denote the distance between two consecutive points p_i and p_{i+1} , and let L_r represent half the robot’s length. Then, we can express the conditions for inserting a new point or merging two points as follows:

$$\begin{cases} \text{Insert new point between } p_i \text{ and } p_{i+1} & \text{if } d_{i,i+1} > L_r \\ \text{Merge } p_i \text{ and } p_{i+1} & \text{if } d_{i,i+1} < L_r \end{cases} \quad (6)$$

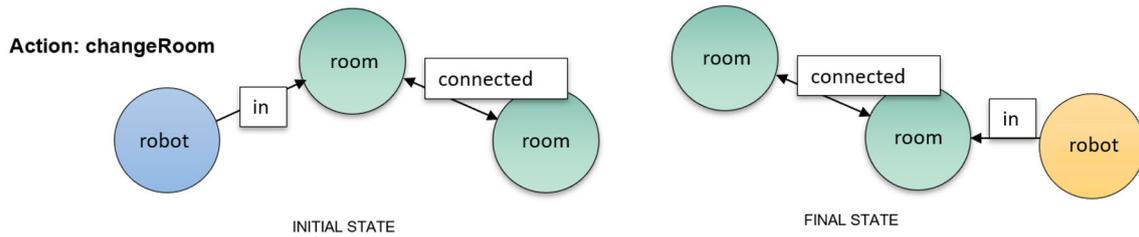


Fig. 8 Original and final states post-execution of the defined actions. In the resultant state, the robot is depicted in the process of transitioning between rooms. This transition mandates the execution of a series of meticulously delineated actions, each residing within the confines of the planning domain

- The following procedure applies an internal force, denoted as f_c , to the elements of the elastic band, counteracting local curvature. This action aims to straighten the elastic band, minimizing the robot's traversal time and energy consumption. The contraction force, f_c , is computed from the neighboring points as follows:

$$f_c = k_c \left(\frac{p_{i-1} - p_i}{\|p_{i-1} - p_i\|} + \frac{p_{i+1} - p_i}{\|p_{i+1} - p_i\|} \right) \quad (7)$$

In the above equation, p_i denotes the location of the i^{th} pass along the route, and k_c represents a constant that characterizes the stiffness of the elastic band. When the three points are collinear, this force becomes null. Figure 9 illustrates the internal forces over the point p_i .

- The subsequent stage involves applying a repulsion force, f_r , to the elastic band, effectively pushing the robot's trajectory away from unmapped obstacles, such as moving objects or people. The magnitude of f_r depends on $D(x, y)$, the shortest distance between point p_i on the path and any obstacles, as determined by the robot's sensors. To compute the maximum variation of $D(x, y)$ concerning point coordinates (x, y) , a discrete Jacobian is employed for each point along the path:

$$\frac{\partial D}{\partial p} = \frac{1}{2\delta} [D(p(x) - \delta x) - D(p + \delta x) D(p - \delta y) - D(p + \delta y)]^T \quad (8)$$

In this expression, D signifies the nearest distance function mentioned earlier, $p(x, y)$ represents the route point, δx and δy are discrete variations in the point's position. The Jacobian is subsequently multiplied by the difference between the highest distance threshold, D_0 , and the current value of $D(x, y)$:

$$f_r = \begin{cases} k_r (D_0 - D) \frac{\partial D}{\partial p} & p < D_0 \\ 0 & p \geq D_0 \end{cases}, \quad (9)$$

being k_r the global repulsion gain [8]. Figure 9 illustrates the repulsive force. By implementing this repulsion force,

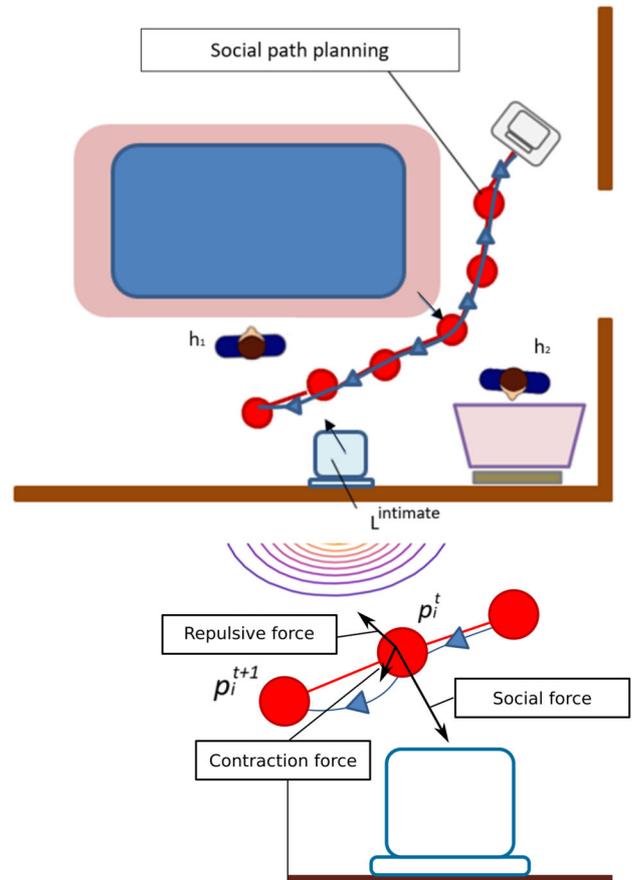


Fig. 9 Initially, the path planner generates a trajectory, illustrated in red, with various forces exhibited by arrows over the point in the trajectory p_i : attraction, repulsive, and social forces. The social route (after the equilibrium of the elastic band) is represented as a continuous blue line, denoted by a triangular symbol

the algorithm ensures the robot's trajectory adapts to avoid unmapped obstacles, promoting safe and efficient navigation through complex environments. However, these objects do not equally impact the robot's navigation, particularly when humans are in the environment. To address this, we require a new component considering human activity in the robot's immediate surroundings.

- The final stage of the process considers the individuals within the environment and applies a social force to the

elastic band, which is contingent upon the distance to social interaction spaces. The algorithm verifies whether the points of the planned trajectory p_i are within any of the social interaction spaces L defined in the social layer of the SNAPE framework, and if so, a social force is created from the affected person or people to alter and smooth the robot's route. Denoted as f_s , this force effectively guides the robot's path away from nearby individuals. The intensity of the social force, f_s^i , is determined by $L_i(x, y)$, defined as the minimum distance from point p_i to the interaction space of person i , L_i^{space} (i.e., $L_i^{intimate}$ if the trajectory is in the intimate area, L_i^{social} if the trajectory is in the social interaction space). For each point p_i , the direction of maximum variation $L_i(x, y)$ concerning the coordinates of the point on the curve L_i^{space} is computed utilizing the discrete Jacobian:

$$\frac{\partial L_i}{\partial p} = \frac{1}{2\delta} [L_i(p - \delta x) - L_i(p + \delta x) \quad L_i(p - \delta y) - L_i(p + \delta y)]^T \quad (10)$$

Subsequently, the difference between the largest distance threshold L_0 and the current value of $L_i(x, y)$ is calculated, and the outcome is multiplied by the Jacobian:

$$f_s^i = \begin{cases} k_s(L_0 - L_i) \frac{\partial L_i}{\partial p} & p < L_0 \\ 0 & p \geq L_0 \end{cases} \quad (11)$$

In this expression, k_s denotes a general social gain and signifies the maximum distance to which the social force is computed. Ultimately, f_s constitutes the sum of all f_s^i associated with individuals in the robot's vicinity. The social force f_s is shown in Fig. 9.

The force f_r adjusts the initial path in response to the current environment, rectifying potential planning errors from imprecise world modeling, loss of robot localization, or unexpected obstacle intersections during path calculation. In contrast, the force f_s swiftly adapts the robot's trajectory to the presence of individuals, considering their social interaction spaces. The forces exerted on each point of the elastic band are proportional to the distance between the robot and the person. The elastic band algorithm employed amends the path to ensure it remains unobstructed by obstacles and people, augmenting the distances to each obstacle and supplying the reactive component necessary for real-time navigation control. Decoupling these two forces facilitates more socially aware robot navigation, with proximity to people determined by the selected values for k_r and k_s .

Ultimately, each point p_i is influenced by a combination of the repulsion force (f_r), the social forces (f_s^i), and the

attraction force (f_c).

$$p_i^{t+1} = p_i^t + f_r + f_c + f_s \quad (12)$$

After a brief duration, the entire path attains an equilibrium point at which all forces are balanced [8]. In Fig. 9, the elastic band is observed in blue, with the subsequent points in the trajectory represented as triangles. The original path is represented in red. In this context, it is important to remark that the forces within the social elastic bands are indeed not in conventional force units but are rather vector quantities that act to modify or adapt the trajectory points, thereby smoothing the path. This process doesn't follow the physical definition of force but adopts a conceptual approach to represent the influence on the trajectory.

5 Experimental Results

This section presents the experimental results of implementing the social elastic band with prediction and anticipation method in both simulated and real scenarios, utilizing the SNAPE framework. We begin by elucidating the computation of optimal force gains. Subsequently, we assess the performance of the proposed method within simulated environments. Lastly, we apply the method to real-world settings and discuss the results.

5.1 Computation of Optimal Force Gains

To achieve optimal performance in real-time, the elastic band agent, which serves to amalgamate global path planning with local path tracking, requires fine-tuning of three free parameters. The optimization of these parameters is essential, as they correspond to the gains that multiply each force acting on the band:

- k_c : This gain corresponds to the global attraction, which scales the attraction force following Eq. 7.
- k_f : As the global repulsion gain, it multiplies the repulsion force as delineated in Eq. 9.
- k_s : The social gain scales the social force as described by Eq. 11.

The setting of the gains, specifically (k_c , k_f , and k_s), is critical to ensure the efficient operation of the social elastic band in dynamic environments. Suppose these gain values are not set accurately. In that case, the band may have difficulty adapting quickly to emerging obstacles or individuals, which can lead to stability problems, especially in scenarios where objects are located nearby. Similarly, inappropriate values negatively influence navigation efficiency and how

they affect times and distances traveled (*i.e.*, energy consumed). In the context of our SNAPE framework, we have explored the influence and implications of these parameter values. First, an examination of the overall attraction gain k_c reveals its fundamental role in guiding the robot to its goal, and its tuning directly influences the robot's adherence to the optimal trajectory, especially in contexts interspersed with obstacles, without considering the people in it. As for the global repulsion gain k_f , its main function is safeguarding the robot from possible collisions with objects, ensuring its respectful distance from obstacles. Very high values move the robot too far away from the path, extending the time to arrive and reducing the efficiency of the navigation algorithm. Conversely, if the values are too small and inadequate, the protection of the robot would be in question. The use of the social force gain k_s , included in our paper, can be likened in terms of effects on navigation to the repulsive force gain k_f , insofar as it moves the robot away from its optimal route in exchange for not disturbing people during navigation.

To determine appropriate gain values, we define a function operating over the robot's path that yields low values when the path is executed optimally. This function, situated in the trajectory space T , consists of four penalty functions, each multiplied by distinct constants:

$$G = \kappa_1 CHC(t_i) + \kappa_2 d_t(t_i) + \kappa_3 \tau(t_i) + \kappa_4 d_h(t_i) \quad (13)$$

In this representation, we used a set of well-established metrics in the scientific community. These included the cumulative heading changes (CHC), the distance traveled (d_t), the navigation time (τ), and the average minimum distance to a human during navigation (d_h). These metrics have been previously reported in the literature [38, 39] and are widely accepted as standard measures for assessing social navigation algorithms.

Formally, CHC is the sum of the robot's orientation changes θ through the trajectory,

$$CHC = \int_0^T d\theta dt \quad (14)$$

d_t signifies the total distance the robot traverses as it navigates through its environment, accounting for all movements and adjustments made along its route to the destination.

$$d_t = \int_0^T \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} dt \quad (15)$$

τ represents the overall time duration taken by the robot to complete its journey along the specified path, whereas d_h denotes the average distance maintained between the robot and the nearest individual encountered throughout its navigation.

gation.

$$d_h = \frac{1}{T} \sum_{i=0}^T d^*(p_i, H), \quad \text{with } d_i^* = \min_k (d(p_i, h_k^i)) \quad (16)$$

where $\{H\}$ are the people visible to the robot from position p_i

In Eq. 13, the coefficients κ_1 , κ_2 , κ_3 , and κ_4 indicate each separate function's contributions to the overall final function. These coefficients determine the weighting and relative significance of the different components within the equation. The gains k_s , k_r , and k_c are intrinsic parameters within each of the functions that constitute G , affecting the behavior and responsiveness of each. They modify how each component of G reacts to the environment and influences the overall trajectory, while the κ_i coefficients determine the relative significance of these reactions within the aggregate operation of the elastic band. Each term of Eq. 13 is normalized to a value between $[0, 1]$. The weights, κ_1 , κ_2 , κ_3 , and κ_4 , are constant throughout all the experiments to maintain a consistent evaluation framework. In descending order of importance, they are configured to emphasize the significance of the metrics d_h , τ , CHC , and d_t . This configuration was chosen to prioritize avoiding immediate hazards and optimizing trajectory while considering human comfort and path length, ensuring a balanced and socially-aware navigation strategy in varying contexts.

To achieve the minimum value of G over a selection of trajectories, we apply a blend of direct search and stochastic gradient approaches. First, we set the values of two gains, leaving one to be adjusted by sampling within a predetermined range. The robot performs 20 distinct routes for each gain value, and we document the resulting path. For each case t_i , we calculate the various functions CHC , d_t , τ , and d_h , which are then incorporated into G_i . We determine the trajectory t^* that minimizes G and use this to determine the third gain. In the next iteration, we set the values of these gains to correspond to the minimum, sample another gain, and perform a new set of paths. This process is repeated until no further improvements are observed in the previous results. We apply this method to two simulated scenarios, the first of which is shown in Fig. 10 (top), where the robot navigates a simple room with a single individual. The planned robot's path, social map, and trajectories for two cases ($k_s = 0$ and $k_s = 27$) are displayed. We have chosen these values to illustrate two representative cases. With k_s equal to zero, we depict what occurs without our new method. With k_s equal to 27, we compel the robot, thanks to the social force, to distance itself from people during navigation, even from the onset of its movement. We present similar results in Fig. 10 (bottom), where the scenario remains the same, but two individuals

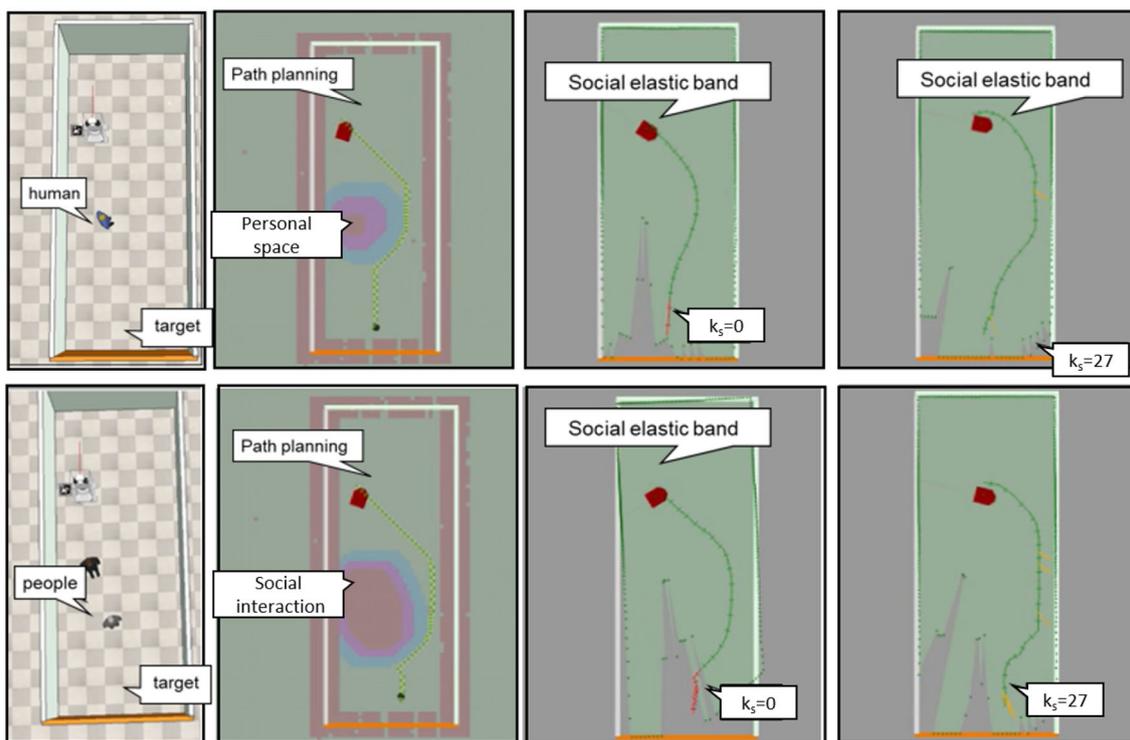


Fig. 10 The figure shown displays the following from left to right: a simulated scenario, the social map, and the planned path. The path optimization is demonstrated using the classic elastic band approach with

$k_s = 0$ and the social elastic band approach with $k_s = 27$. For $k_s = 27$, the path comes dangerously close to the wall, which is also not socially acceptable. The optimization results are reflected in the Table 1

Table 1 Principal values and optimized gains utilized in the experiments

Parameter	Value	Description
k_c	10	Global attraction gain
k_f	2	Global repulsion gain
k_s	10	Social force gain
v_{fmax}	800 mm/s	Maximum forward robot's velocity
D_0	1000 mm	Highest distance threshold (to object)
L_0	2000 mm	Highest distance threshold (to humans)

interact with one another- Additional results for different values of k_s can be observed in the accompanying videos.²

In order to provide a clear and quick reference to the principal values and gains utilized within our experiments, we present Table 1 below. This table encapsulates not only the optimized gains, k_c , k_f , and k_s , derived from the iterative optimization procedure detailed earlier, but also other values and parameters crucial to the functioning and outcomes of our experiments. Each value and parameter have been tuned and chosen to ensure the robustness and relevance of our experimental findings.

² Link to the sample of the tests: <https://cloudiepcc.unex.es/index.php/sjYxtmxfwDBbyPcR>.

5.2 Simulated Scenarios

To validate the real-time path optimization algorithm, we conducted experiments in simulated environments where people may be standing or moving in close proximity. This allowed us to evaluate different scenarios and identify potential problems that may arise during real-world testing. The experiments were conducted on a personal computer with an Intel Core i7 processor, 8GB of DDR3 RAM, and Ubuntu GNU/Linux 20.10 operating system.

To evaluate the effectiveness of the proposed approach, we used the same set of social metrics described in 5.1. Besides, we add the personal space intrusions (Ψ) and the minimal distance to people d_1 and d_2 . These new metrics have also been previously reported in [38, 39].

In this scenario, a 65-square-meter apartment with a living room, open kitchen, and corridor is simulated using the Vrep simulator. The simulated environment includes multiple RGBD cameras and a social robot with an omnidirectional base equipped with an RGBD camera. All scenarios include individuals in the vicinity of the robot. These individuals are in fixed positions to evaluate the effect of the combination of forces on the robot's path adaptation. The static objects that include social interaction spaces with people are the table and the refrigerator, as can be seen in

Table 2 Navigation results for the experiment shown in Fig. 11

Parameter	Simulated environment					
	Test 1		Test 2		Test 3	
	Social elastic band	Elastic band (EB)	Social EB	EB	Social EB	EB
d_t (mm)	1757.13	1428.02	1253.30	1199.52	1529.17	1651.62
τ (s)	9.07	10.05	13.21	13.52	10.76	9.92
CHC	4.68	3.77	6.22	5.63	4.18	5.22
d_1 (mm)	1416.77	1355.2	1016.65	1009.63	1416.78	985.21
d_2 (mm)	1690.81	671.08	871.56	734.75	1690.81	1949.39
Ψ (Intimate) (%)	0	0	0.00	0.00	0.00	0.00
Ψ (Personal) (%)	0	5.45	0.00	3.15	0.00	0.44
Ψ (Social) (%)	0	0	0.00	0.00	0.00	0.00
Ψ (Public) (%)	100	94.55	100.00	96.85	100.00	99.56

the social map. In all the experiments, two people are in the robot's surroundings. The Cortex architecture is utilized for people detection and tracking, estimating future poses, and the classical version of the SNAPE framework, which employs the Elastic Band algorithm, is employed for navigation.

5.2.1 Scenario with Static People

Two tests were performed for scenarios with static people. The results of the two tests conducted, as indicated in Table 2, suggest that the proposed social optimization approach ($k_s = 10$) outperforms the classical Elastic Band algorithm ($k_s = 0$) in terms of human comfort and safety. Figure 11 pro-

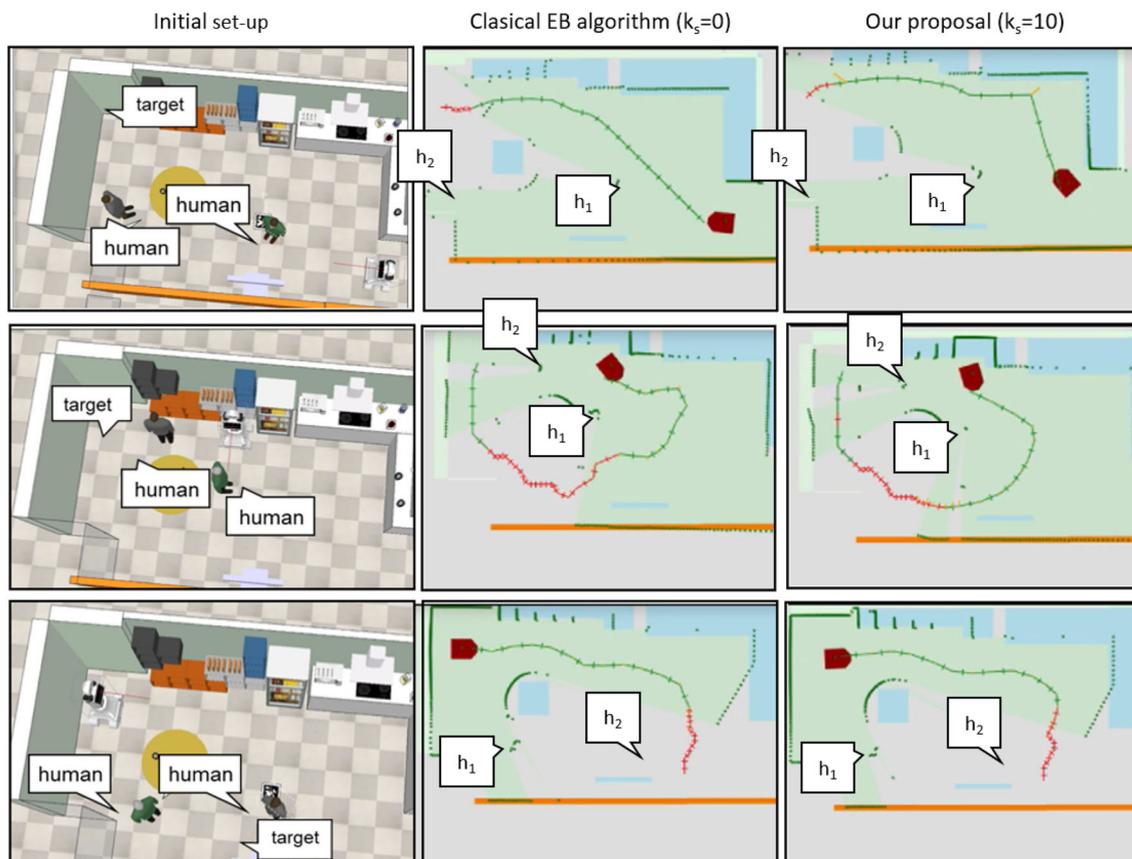


Fig. 11 Three different simulated scenarios. From left to right: initial set-up, path optimized by the classical elastic band algorithm, and, finally, the trajectory optimization with the social elastic band

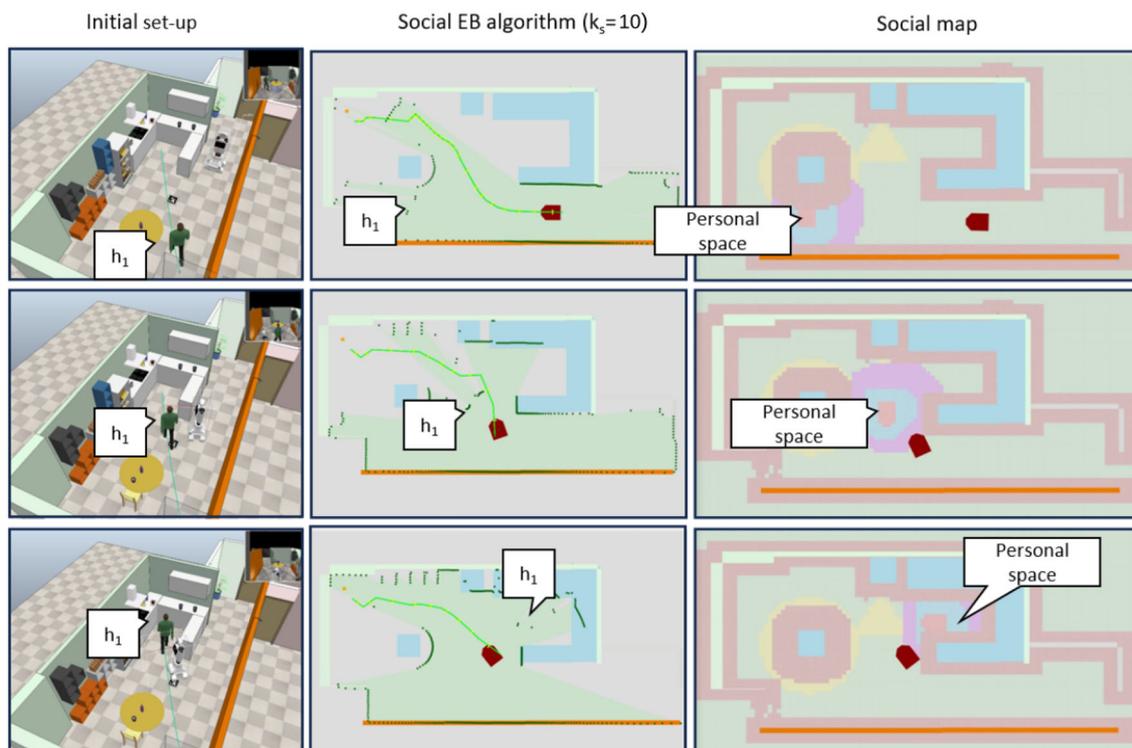


Fig. 12 Navigation with a person in motion. From up to down: initial set-up, path optimized due to social force when the person approaches the path, and, finally, the trajectory returns to the initial shape when the

person disappears. From left to right: initial set-up, path optimized by the social elastic band algorithm proposed in this paper, and the social map at this instant

vides a visual representation of the scenario, depicting the initial setup, the optimized path using the classical Elastic Band algorithm, and the optimized trajectory using the social Elastic Band. The non-social optimization resulted in shorter distances traveled by the robot in a similar period; however, it resulted in longer invasions of personal spaces, as indicated by the high value of $\Psi(Personal)$. In contrast, the social optimization approach resulted in minimal distances d_1 or d_2 , avoiding personal space invasions and ensuring a higher degree of human comfort.

Furthermore, the cumulative heading changes (CHC) and the navigation time (τ) indicate that the social optimization approach prioritizes human comfort and safety and achieves efficient navigation. The experiment results demonstrate the proposed approach's effectiveness in improving the performance of social navigation algorithms, making them safer and more efficient for robot navigation in environments with people. From the previous results, it is evident how combining a global planner and the social elastic band allows the robot to maintain prudent distances from people without invading their personal spaces, considering these people independent of the scene's objects. In the first test, the robot, using the proposed navigation framework, moves away from the person and closer to the furniture. If this social force is

not considered, the approach to the person would be greater, provoking fewer social situations. The same occurs in the rest of the experiments. The video of the scenario provides a visual representation of the results and further supports these conclusions.³

5.2.2 Scenario with People in Motion

Two experiments have been performed in which people move around the robot. In the first experiment, shown in Fig. 12, a person walks from one end of the living room to the kitchen. In the second experiment, shown in Fig. 14, in addition to this person, a person is walking in the hallway. In these experiments, we aim to validate the anticipation proposed in our paper by estimating the person's position at a future point in time ($t = 2s$) and reducing the speed if there is a risk of collision. To observe the behavior of the robot in environments with moving people, the speed assigned to the robot by the controller, together with the minimum distance to people d_{min} in the surrounding area, have been analyzed in addition

³ link to the video <https://cloudieccc.unex.es/index.php/s/jYxtmxfwDBbyPcR>.

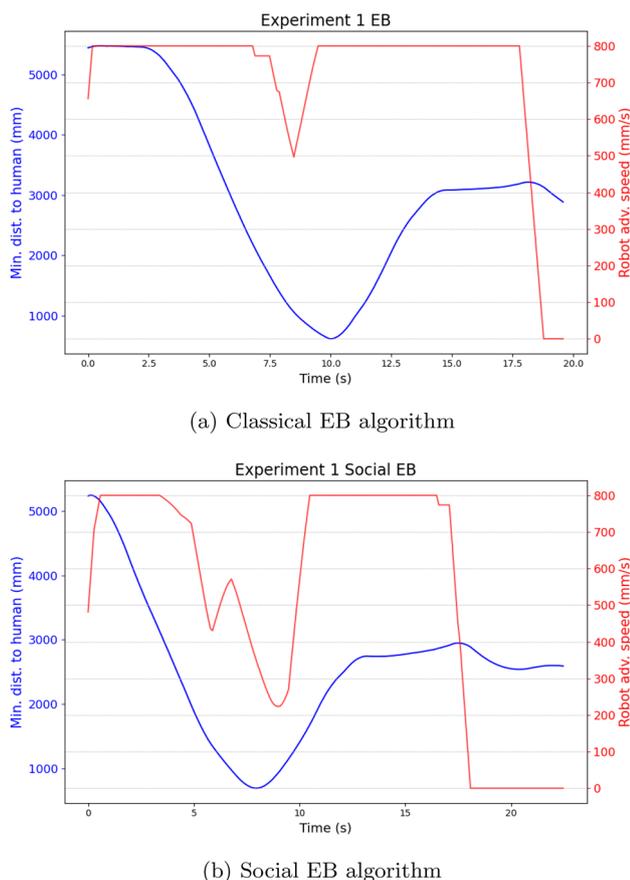


Fig. 13 Results of Experiment 1. Evolution of the robot's speed throughout the experiment (in red). The graph also depicts the distance between the robot and the person in blue

to the Ψ values:

$$d_{min} = \min_{p \in H} \|r_0 - p\| \quad (17)$$

where r_0 is the robot pose.

Figures 13 and 15 show the experiment's results. The evolution of the robot's speed is analyzed, depicted in red in the graph, to understand the dynamic adjustments made by the robot during the experiment. Additionally, the graph displays the distance between the robot and the estimated position of the person in blue, allowing for a comprehensive view of the relationship between speed adjustments and proximity to humans. A comparative analysis was conducted between the Classic Elastic Band (EB) algorithm and the new Social EB algorithm proposed in this paper. Figure 13a shows that the robot keeps the speed constant during most of the experiment. It is possible to observe how this decrease in velocity is caused by the robot colliding briefly with the human by looking at the attached video. Conversely, Fig. 13b shows how the velocity adjusts accordingly as the distance to the nearest person decreases, in the scenario shown in Fig. 12. Thus, the

velocity value decreases to approximately 200 mm/s, allowing the person to navigate safely.

In terms of personal space intrusion, it is possible to observe in the Table 3 that the use of the Social EB together with the speed setting is more respectful since the invasion of the social space Ψ (Social) is only 2.22% of the time. On the other hand, it is observed that when only EB is used, the social space is invaded for 5.05% of the time, and even the personal space ψ (Personal) is invaded for 0.87% of the navigation time.

A similar behaviour to that obtained in Experiment 1, shown in Fig. 12, can be observed in the second experiment, shown in Fig. 14. Similarly, when using only the EB, the robot's speed is only reduced when it collides with a person. Regarding the case of using Social EB, it can be observed that the speed is slightly reduced without compromising the person's safety by not predicting a critical position for the first person. In the case of the second person, predicting his trajectory allows the robot to significantly reduce speed, allowing the person to navigate without compromising safety.

With regard to the invasion of personal spaces, it is observed that, as in the previous experiment, the combination of SEB with the modification of the speed using prediction reduces the intrusion of the robot into the personal spaces of the people. Specifically, in the case of the SEB, the value of the percentage of Ψ (Personal) along the path is 3.94%, while in the case of the EB, it is 6.33%.

5.3 Real Scenarios

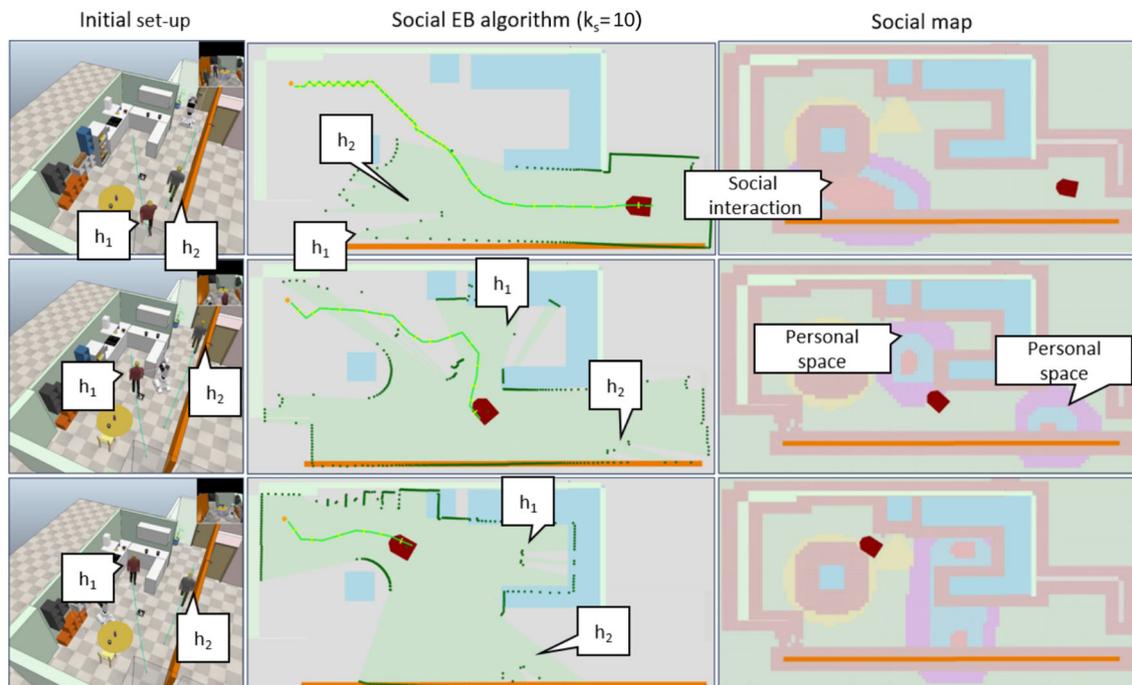
To evaluate the effectiveness of our approach in real-world scenarios, we conducted experiments using the semi-humanoid robot Viriato. With the ability to move in all directions, Viriato stands at approximately 1.7 m tall and is equipped with various cameras to facilitate navigation (SLAM) and interaction with objects and people. In addition, a laser sensor provides continuous environmental feedback.

The experiment was conducted in a 65 m² flat with two rooms, with the kitchen-living room configuration used for this test. The other room remained empty and had no impact on our experimental design. In both the experiments conducted in simulated environments and real-world tests, the objects inserted into the social layer of the SNAPE framework consist of a table and a refrigerator. Three RGBD cameras were installed in the experiment room, connected to the NVIDIA Jetson Nano development kit, to enhance the detection and tracking of the positions of different individuals in the environment. In this instance, we have considered two scenarios where individuals are in static positions, carrying out tasks typical of an elderly care center.

The CORTEX structure employed in these experiments consists of various agents presented in our article, distributed across multiple computers running the Linux Ubuntu 20.04

Table 3 Person spaces intrusions for the experiments shown from Figs. 13, 14 and 15

Parameter	Simulated Environment with people in motion			
	Test 1		Test 2	
	Social EB	EB	Social EB	EB
Ψ (Intimate) (%)	0,00	0,00	0,00	0,00
Ψ (Personal) (%)	0,00	0.874636	3.94238	6.32506
Ψ (Social) (%)	2.21654	5.05345	5.38287	5.76461
Ψ (Public) (%)	97.7835	94.0719	90.6748	87.9103

**Fig. 14** Navigation with two people in motion. From up to down: initial set-up, path optimized due to social force when the person(s) approaches the path, and, finally, the trajectory returns to the initial shape when peo-

ple disappear. From left to right: initial set-up, path optimized by the social elastic band algorithm proposed in this paper, and the social map at this instant

distribution. The RoboComp framework [40] must also be installed for the system to function properly. RoboComp is an open-source framework designed to develop and integrate various robotic components seamlessly. In our proposal, RoboComp is integral, providing the necessary tools and environment for the development, deployment, and execution of the various robotic components and modules, enabling efficient communication and interaction between them. The environment used to perform the tests is shown in Fig. 16.

During the robot's navigation through the environment, it must be mindful of human presence to avoid causing disturbances. To evaluate the effectiveness of our proposed algorithm, we compared it to the elastic band optimization approach. The results are depicted in Fig. 16, where we

observe that our algorithm causes the robot to move further away from people, promoting more socially acceptable behavior.

Table 4 presents the metrics for assessing the algorithm's performance. The results showed that our algorithm promotes more socially acceptable behavior in the robot according to the metrics used in our comparative study. Specifically, we observed that the robot moved further away from people and avoided personal space violations. Moreover, our algorithm demonstrated an ability to navigate in a way that was both efficient and socially aware despite traveling greater distances and taking slightly longer to complete the task. Our experimental results indicate that our proposed algorithm could effectively solve social navigation tasks.

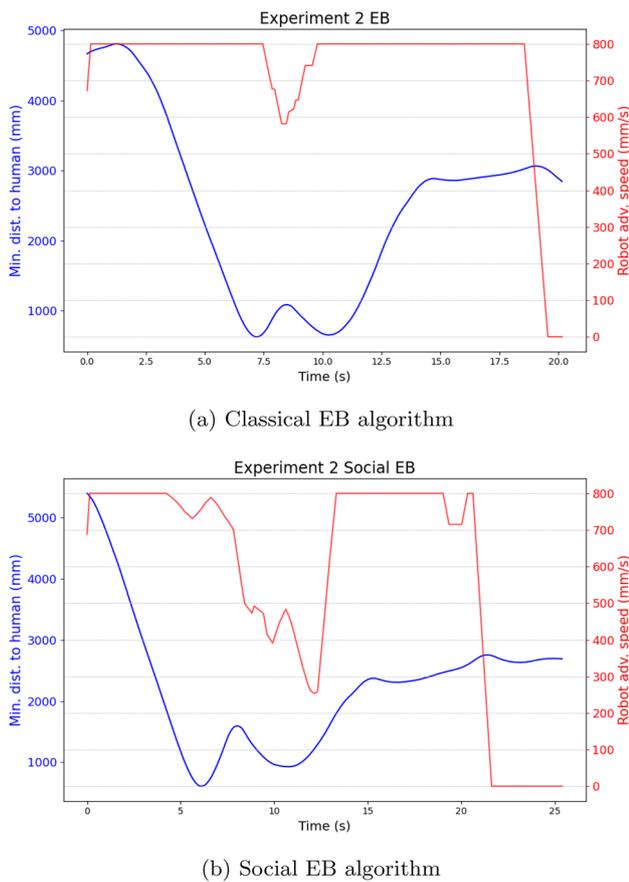


Fig. 15 Experiment 2 results. Evolution of the robot's speed throughout the experiment (in red). The graph also depicts the distance between the robot and the person in blue

5.4 Algorithm Performance Analysis

Pursuing a socially-aware navigation strategy necessitates functional robustness and computational efficiency, especially in real-time applications where system resources are often at a premium. Our SNAPE framework mediates socially adept navigation without imposing untenable computational demands. The subsequent data elucidates our algorithm's computational and memory usage and its constituent modules. We present a summation of relevant metrics extracted during our algorithm's operation in Table 5. These metrics encompass CPU and RAM usage both when the software components are idle, as well as during robot navigation; and update frequencies, depicting each software agent's computational comportment and overall system demands. The results do not show a significant increase between the two operating states. Only those responsible for mission planning and development are slightly affected.

The hardware described in Sect. 5.2 is the hardware used to obtain the values in Table 5. The refresh rates of the agents involved are closely related to the data acquisition rate of the sensors implemented on the robot and in the scene, and to the size of the grid. Changing the hardware does not necessarily cause the refresh rate to decrease, unless the computational characteristics of the hardware cause a bottleneck. The agents with the highest computational load are those involved in performing operations on the grid and therefore most susceptible to refresh rate drops, such as the Human Social Spaces and Social Mapping agents, which are responsible for the generation of social spaces and the dynamic updating of the grid.

During the development process, we have prioritised optimising the data acquisition and processing procedures to

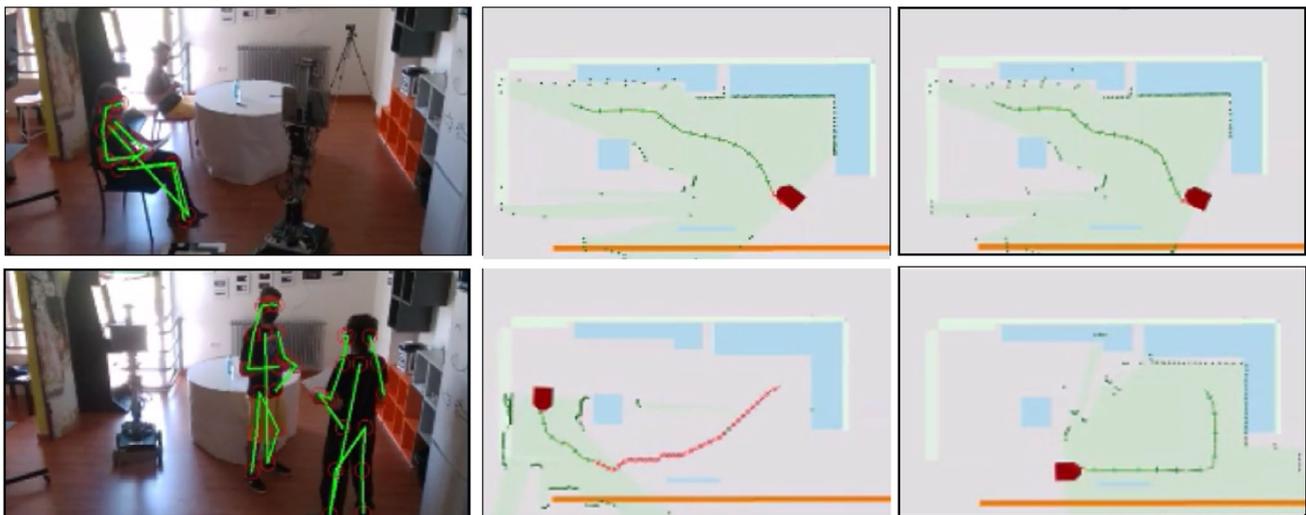


Fig. 16 Two real scenarios for the experiments From left to right: initial set-up, path optimized by the classical elastic band algorithm, and, finally, the trajectory optimization with the social elastic band

Table 4 Navigation results for the experiment shown in Fig. 16

Parameter	Real environment			
	Test 1		Test 2	
	Social Elastic Band	Elastic Band (EB)	Social EB	EB
d_t (mm)	1689.50	1393.38	1983.15	1774.25
τ (s)	7.16	5.70	8.50	8.95
CHC	2.22	2.73	3.69	3.05
d_1 (mm)	1070.47	852.28	1655.50	1469.45
d_2 (mm)	1965.80	2058.88	1622.13	1400.83
Ψ (Intimate) (%)	0.00	0.00	0.00	0.00
Ψ (Personal) (%)	0.00	13.95	0.00	2.90
Ψ (Social) (%)	0.00	0.00	0.00	6.05
Ψ (Public) (%)	100.00	86.05	100.00	91.05

Table 5 Computational and memory usage of algorithm modules

Software agents (CORTEX architecture)	CPU idle (%)	RAM idle (MB)	CPU work (%)	RAM work (MB)	Update frequency (Hz)
Human social spaces	103	187.8	106	187.8	16
Viriato DSR (robot data obtaining and people detection and tracking)	36.00	147.71	44	158.61	20
Global path planner	50	113.61	74	118.55	20
Social elastic band	32	91.85	38	101.83	17
Controller	24	87.44	34	90.43	10
Social mapping	108	106.98	106.25	130.04	16

minimise associated latency. To ensure the most up-to-date information possible, we have aimed to keep the agents responsible for robot control working at a rate of 10 Hz, while the remaining agents work at higher rates. In the experiments, a grid cell size of 100 mm was used and a minimum time of 62 ms was achieved for the grid updates, compared to the 100 ms period of the robot control.

6 Conclusions and Future Works

This paper presents a novel real-time path trajectory optimization algorithm for socially aware robot navigation, incorporating prediction and anticipation for reasoning in robot navigation. The algorithm is based on the social elastic band concept, which distinguishes between static objects and human presence, allowing for the definition of personal spaces and their relationship to the elastic band. The algorithm rapidly adapts to environmental changes without causing disturbance, generating socially accepted paths and adapting speed to ensure social acceptance during human-robot interaction.

The proposed algorithm has been integrated into a social navigation framework and tested and validated through simulations and real-world experiments in various environments. The experimental results demonstrate that the algorithm effectively could maintain socially acceptable behavior while adapting its motion to people's poses. The algorithm's ability to anticipate and predict changes ensures efficient and socially aware navigation, paving the way for the seamless integration of social robots into human environments.

One of the most prominent challenges arises when the robot navigates through densely populated areas, where the wide variety of social interactions and the unpredictable nature of human movements can impose significant computational demands on the algorithm, potentially hindering real-time responsiveness and navigation path optimization. In addition, the approach may encounter difficulties in situations where human behaviors diverge from predicted patterns, requiring new advances in anticipatory algorithms and predictive models to improve the robustness and reliability of the social rubber band in dynamically evolving social contexts.

Building upon these points, the mobility and perception capabilities disparities among various robotic forms further magnify the challenge. While the social elastic band mechanism has shown practical applicability in certain contexts, its universal application across different robotic systems, from differential to omnidirectional mobility types, invites further explorative research and algorithm refinement. Notably, a robot's physical and perceptual attributes, whether related to its size, sensor positioning, or mobility mechanism, are not merely technical details but crucial facets that determine its interactive and navigational capability within social environments. As such, a robot's specific characteristics may demand bespoke modifications to the algorithm to preserve socially respectful and contextually aware navigational behaviors. The development and validation of the proposed approach in these contexts could hence foster a more universally applicable and robust socially aware navigation strategy, enabling robots to navigate with social adeptness across a broader array of scenarios and robotic platforms.

The proposed algorithm can be further improved by incorporating more advanced prediction and anticipation techniques, such as machine learning and deep reinforcement learning. Additionally, the algorithm's performance can be tested in more complex environments, such as those with crowds. Furthermore, we anticipate conducting comprehensive user studies to validate our method's social acceptance and perceived sociability against various other approaches, aiming to substantiate our claim with evidence from human participants. Finally, the proposed algorithm can be tested on different types of robots to evaluate its effectiveness and generalizability.

Acknowledgements This work has been partially funded by the PDC2022-133597-C41 project and TED2021-131739-C22 R+D+i projects, supported by Spanish MICIU/AEI/10.13039/501100011033 and the European Union's "NextGenerationEU"/PRTR, by the PID2022-137344OB-C31 R+D+i project, supported by Spanish MICIU/AEI/10.13039/501100011033 "FEDER Una manera de hacer Europa", "FEDER /UE", and by the FEDER Project 0124_EUROAGE_MAS_4_E (2021–2027 POCTEP Program).

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest

Consent to publish The authors acknowledge that the individuals involved in the tests are coworkers who provided informed consent for participation in this study and for the publication of the images in Fig. 16.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as

long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Hall E (1968) Proxemics. *Curr Anthropol* 9(2–3):83–108
- Vega A, Manso LJ, Bustos P, Núñez P, Macharet DG (2019) Socially aware robot navigation system in human-populated and interactive environments based on an adaptive spatial density function and space affordances. *Pattern Recogn Lett* 1:72–84
- Rios-Martinez J, Spalanzani A, Laugier C (2015) From proxemics theory to socially-aware navigation: a survey. *Int J Soc Robot* 7(2):137–153
- Charalampous K, Kostavelis I, Gasteratos A (2017) Recent trends in social aware robot navigation: a survey. *Robot Autonom Syst* 93:85–104
- Tung TX, Ngo TD (2018) Socially aware robot navigation using deep reinforcement learning. In: 2018 IEEE Canadian conference on electrical and computer engineering (CCECE), Quebec City, QC
- Möller R, Furnari A, Battiato S, Härmä A, Maria-Farinella G (2021) A survey on human-aware robot navigation. *Robot Auton Syst* 145:103837
- Vega A, Gondkar R, Manso L, Núñez P (2021) Towards efficient human–robot cooperation for socially-aware robot navigation in human-populated environments: the snape framework. In: IEEE International conference on robotics and automation
- Quinlan S, Khatib O (1993) Elastic bands: connecting path planning and control. In: Proceedings IEEE international conference on robotics and automation, Atlanta, GA, USA. vol 2, pp 802–807
- Althaus P, Ishiguro H, Kanda T, Miyashita T, Christensen HI (2004) Navigation for human–robot interaction tasks. In: IEEE international conference on robotics and automation, vol 1, pp 1894–1899
- Kruse T, Pandey A, Alami R, Kirsch A (2013) Human-aware robot navigation: a survey. *Robot Auton Syst* 61(12):1726–1743
- Calderita LV, Vega A, Bustos P, Núñez P (2021) A new human-aware robot navigation framework based on time-dependent social interaction spaces: an application to assistive robots in caregiving centers. *Robot Auton Syst* 145:103873
- Thoa T, Copot C, De Keyse R (2016) Heuristic approaches in robot path planning: a survey. *Robot Auton Syst* 86:13–28
- Costa M, Silva M (2019) A survey on path planning algorithms for mobile robots. In: 19th IEEE International conference on autonomous robot systems and competitions
- Takayama L, Pantofaru C (2009) Influences on proxemic behaviors in human–robot interaction. In: 2009 IEEE/RSJ international conference on intelligent robots and systems, pp 5495–5502
- Sisbot EA, Marin-Urias LF, Alami R, Simeon T (2007) A human aware mobile robot motion planner. *IEEE Trans Robot* 23(5):874–883
- Kim B, Pineau J (2016) Socially adaptive path planning in human environments using inverse reinforcement learning. *Int J Soc Robot* 8(1):51–66
- Trautman P, Krause A (2015) Unfreezing the robot: navigation in dense, interacting crowds. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 797–803

18. Fox D, Burgard W, Thrun S (1997) The dynamic window approach to collision avoidance. *IEEE Robot Autom* 4(1):23–33
19. Khatib O (1986) Real-time obstacle avoidance for manipulators and mobile robots. *Int J Robot Res* 5(1):90–98
20. Berg J, Lin M, Manocha D (2008) Reciprocal velocity obstacles for real-time multi-agent navigation. In: *IEEE international conference on robotics and automation*
21. Helbing D, Molnár P (1995) Social force model for pedestrian dynamics. *Phys Rev E* 51:4282–4286
22. Cai K, Wang C, Cheng J, Song S, Silva C, Meng M (2020) Mobile robot path planning in dynamic environments: a survey. *arXiv preprint arXiv:2006.14195*
23. Wang Y, Li J, Wang Y, Li J (2020) Predicting human navigation goals based on bayesian inference and activity regions. *Robot Auton Syst* 135:103–111
24. Zhang Y, Liu Y, Wang Y, Li J (2022) Learning personalized human-aware robot navigation using virtual reality. *arXiv preprint arXiv:2203.14741*
25. Samsani SS, Muhammad MS (2021) Socially compliant robot navigation in crowded environment by human behavior resemblance using deep reinforcement learning. *IEEE Robot Autom Lett* 6(3):5223–5230
26. Kobayashi Y et al (2022) Robot navigation based on predicting of human interaction and its reproducible evaluation in a densely crowded environment. *Int J Soc Robot* 14(2):373–387
27. Peltzer O, Asmar DM, Schwager M, Kochenderfer MJ (2023) Incorporating human path preferences in robot navigation with minimal interventions. *arXiv preprint arXiv:2303.03530*
28. Kitagawa R, Liu Y, Kanda T (2021) Human-inspired motion planning for omnidirectional social robots. In: *Proceedings of the 2021 ACM/IEEE international conference on human-robot interaction*, pp 34–42. *ACM/IEEE*
29. Singamaneni PT, Favier A, Alami R (2022) Watch out! there may be a human. Addressing invisible humans in social navigation. In: *2022 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp 11344–11351. *IEEE*
30. Kollmitz M, Koller T, Boedecker J, Burgard W (2020) Learning human-aware robot navigation from physical interaction via inverse reinforcement learning. In: *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. *IEEE*, pp 11025–11031
31. Melo F, Moreno P (2022) Socially reactive navigation models for mobile robots. In: *2022 IEEE international conference on autonomous robot systems and competitions (ICARSC)*, pp 91–97. <https://doi.org/10.1109/ICARSC55462.2022.9784789>
32. Ferrer G, Sanfeliu A (2019) Anticipative kinodynamic planning: multi-objective robot navigation in urban and dynamic environments. *Auton Robots* 43(6):1473–1488
33. Repiso E, Garrell A, Sanfeliu A (2022) Adaptive social planner to accompany people in real-life dynamic environments. *Int J Soc Robot*. <https://doi.org/10.1007/s12369-022-00937-3>
34. Bustos P, Manso L, Bandera JAB, García-Varea I, Martínez-Gómez J (2019) The cortex cognitive robotics architecture: use cases. *Cognit Syst Res* 55:107–123. <https://doi.org/10.1016/j.cogsys.2019.01.003>
35. Bustos García P, García JC, Cintas Peña R, Martinena Guerrero E, Bachiller Burgos P, Núñez Trujillo P, Bandera A (2021) DSRD: a proposal for a low-latency, distributed working memory for cortex. In: Bergasa LM, Ocaña M, Barea R, López-Guillén E, Revenga P (eds) *Advances in physical agents II*. Springer, Cham, pp 109–122
36. Wang C-Y, Bochkovskiy A, Liao H-YM (2022) YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv:2207.02696*
37. Zhang Y, Sun P, Jiang Y, Yu D, Yuan Z, Luo P, Liu W, Wang X (2021) Bytetrack: multi-object tracking by associating every detection box. *CoRR arXiv:2110.06864*
38. Okal B, Arras K (2016) Learning socially normative robot navigation behaviors with bayesian inverse reinforcement learning. In: *IEEE international conference on robotics and automation*, pp. 2889–2895
39. Kostavelis I (2017) Robot behavioral mapping: a representation that consolidates the human–robot coexistence. *Robot Autom Eng* 1:1–3
40. Manso LJPB, Bustos P, Núñez P, Cintas R, Calderita LV (2010) Robocomp: a tool-based robotics framework. In: *Simulation, modeling and programming in autonomous robots*, vol 6472, pp 251–262

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