Noname manuscript No.
(will be inserted by the editor)

A new human-aware robot navigation framework based on time-dependent social interaction spaces: an application to assistive robots in caregiving centers

L. V. Calderita · A. Vega · P. Bustos · P. Núñez

Received: date / Accepted: date

Abstract One of the most critical problems to be addressed by future generation socially-assistive robots working in semi-organized social environments, such as shopping centers, nursing homes, airports, hospitals, or assisted living centers, is the capability of human-aware navigation. Autonomous navigation in a complex environment with people, staff with different roles, timetables, and restrictions to access, among others, requires adapting to socially accepted rules. Consequently, the path-planner must consider concepts related to proxemics and personal spaces of interaction that include human-human, human-robot, or human-object combinations. Likewise, the speed of approaching people, both to initiate communication or to navigate nearby, must be adapted to social conventions. Some of these situations have already been studied in the literature with varying degrees of success. However, the concept of time dependency or *chronemics* in the robot social navigation has been poorly explored. Current algorithms do not take into account the social complexity of real environments and their relationship with the time of day or the activities performed in these scenarios. This article presents a new framework for robot social navigation in human environments, introducing the concept of time-dependent social mapping. The main novelty is that the social route planned by the robot considers variables that depend on the time and the scheduled center activities. The article describes how the areas of interaction vary over time and how they affect human-aware navigation. To this end, the proposed navigation stack defines a new function for time-dependent social interaction space that takes continuous values and is configurable by the center's staff. The global path-planner uses this function to choose dynamically a socially accepted path to the target. Then, the framework uses an elastic band path optimizer as a local planner, adapting the robot's navigation to possible changes during the trajectory. Several use cases in simulated caregiving centers have been explored to validate the robot's social navigation improvements using these temporal variables.

L. V. Calderita
Department of Mechanical Engineering, Computer, and Aerospace Science, University of Leon, Spain.
E-mail: lv.calderita@unileon.es

A. Vega-Magro
Robotics and Artificial Vision Lab. RoboLab Group, University of Extremadura, Spain.
E-mail: avegamag@alumnos.unex.es

P. Bustos
Robotics and Artificial Vision Lab. RoboLab Group, University of Extremadura, Spain.
E-mail: pbustos@unex.es

P. Núñez
Robotics and Artificial Vision Lab. RoboLab Group, University of Extremadura, Spain.
E-mail: pnuntru@unex.es

Keywords social robot navigation · human-aware navigation · social mapping

1 Introduction

The aging population is a demographic problem that modern societies will have to face in the coming decades. Governments and institution's decisions vary, ranging from basic social policies to investments in technology that will facilitate an improvement in the elderly quality of life. In this context, the use of social robots in assisted living environments will become a reality in the next years. At present, there are many real situations in which collaboration between humans and robots could improve older adults' care. In the specific case of a care facility, for example, the possible scenarios include physical or cognitive rehabilitation activities, social-emotional stimulation, the accompaniment side-by-side of the older person while walking, or clinical staff support (1). In all these situations, the robots must carry out a series of specific complex tasks, adapting their behavior to people and the environment.

Social assistive robots, usually known as SARs, are no longer just autonomous platforms. Today they require that their behaviors be socially accepted. Thus, these robots do not work alone and are usually integrated into smart environments equipped with IoT technology and complex perceptual architectures to make socially conscious decisions in real-time (2). These architectures model the perceived information about humans, robots, and the environment to reason, interpret human intentions and perform autonomously. The application of fuzzy logic in Human-Robot Interaction could contribute to deal with the challenge of infer high-level human intention or activities and improve robot's social behavior in these complex environments (3)(4)(5). One of these essential behaviors that the robot must be endowed with is navigating socially in an environment with people, which includes planning and following paths in a social-aware fashion as an essential task to achieve social acceptability (6).

Most works of literature are based on the concept of *proxemics*. Proxemics is defined as the study of humankind's perception and use of space (7). The robot has a socially accepted behavior if it can move through the environment without disturbing people's personal spaces or interrupting their interaction with other people or objects in the environment. Social path planning in these scenarios is usually solved following *social mapping* strategies that map regions of the environment in which the robot should not navigate (8). Social mapping extends metric and semantic maps by including social information of the environment. Consider the scenario depicted in Fig. 1a: the path planned by the robot in a caregiving environment takes into account people who might be interacting with some of the objects. The interaction space associated with an object is known as *affordance* (9). In Fig. 1a, around the object *table*, its interaction space has been drawn in red during a group therapy session with older people. Consequently, the robot avoids this area in its trajectory. However, the object's interaction space should not be static but should vary over time. In the above illustrative example, the object *table* is not always being used. Indeed, its use depends on the therapeutic session scheduled. Fig. 1b represents another situation, where the robot plans a different trajectory, based on the activities schedule, without the risk of invading the object interaction space.

This dependence over time in social mapping is poorly studied in the robotic literature. However, the time dependence in interactions between people is known in other disciplines as *chronemics* (10). According to this theory, time influences human interactions, and the use of time can affect movements, behaviors, or even, how long people are willing to listen. Hence, a social robot must be able to take time into account when planning its trajectories in the environment with people. For example, if an interaction between one person and another has just begun, an interruption due to a robot's navigation is perceived as worse than if the conversation takes longer. In the same way, if an activity programmed in the agenda advertises the use of an object by a group of people, time defines the occupation of space and how the robot's navigation could disturb the activity development. These situations are every day in museums, educational centers, shopping centers, or

convention centers, where there is prior planning of the activities that will use the facilities. The example described in Fig. 1 can be extended to other real scenarios in care facilities where activities are governed by schedules established by the clinical or administrative staff.

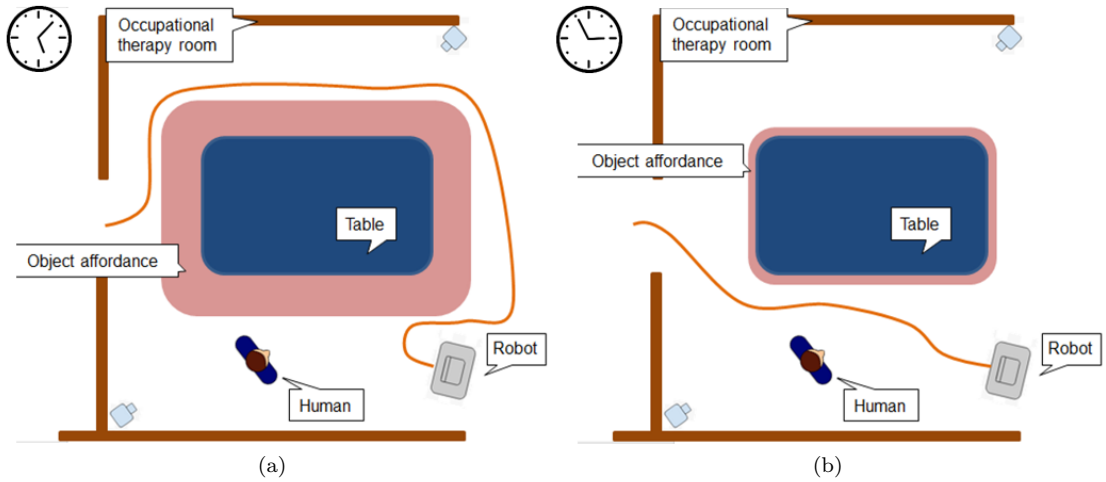


Fig. 1: Two different everyday scenarios in a caregiving center: a) The social path planned by the robot takes into account that there is scheduled a therapy in the occupational therapy room; b) Unlike the previous case, in this new scenario, there is not scheduled therapies in this room.

Unconsciously, we use the time in our daily lives when we think about where to go. Therefore, an interesting topic is how to include the concept of time in the navigation of a social robot. Currently, some solutions predict the position of people in the environment or whether they are using an object at a specific instant of time. However, this is not enough for schedule-driven caregiving centers, where it is complex to distinguish between an informal meeting in a room or group therapy according to the information acquired by sensors. In the first situation, if the person enters the room, it would not violate established social norms. On the other hand, in group therapy, the same interruption would disturb the session. Likewise, it is easy to know if a TV is on or not, but it is not the same if the TV is on for a playful purpose or if it is being used in a therapeutic session, for example, in a session based on Serious Games. If the robot knows the center's activity schedule, it can plan trajectories that do not disturb the elderly and caregivers. Therefore, if its path planner includes time, the path could be easily adapted and achieve a higher degree of social acceptance.

As can be appreciated in the problem definition, described in Fig. 1, robot navigation in an environment with humans is extremely complex. In a simple case, it requires people's detection and tracking in the environment, modeling their personal interaction spaces and planning the path which, besides, must be dynamic and adapt to changes. In more complex situations, such as those described in this article, it is necessary to know which objects and people interact with and how this relates to the caregiving center's activities scheduled. All the above implies a complex and global perception system, not only dependent on the sensors with which the robot is equipped. Most current systems use environments with a sufficient number of deployed sensors (cameras, microphones, etc.), which integrate all the information in a representation of the common knowledge. In this context, the cognitive architecture CORTEX used in this paper and described in (11) is based on a collection of agents (*i.e.*, semi-autonomous functional units that collaborate using a common representation in their pursuit of a common goal) that can operate anywhere in the deliberative-reactive spectrum.

In this architecture, there are navigation, perceptual, and human-robot interaction agents, among others, thereby facilitating the combined use of classic navigation and social rules. In addition, the CORTEX architecture includes planning agents that, based on a specific domain, generate various actions that the robot executes sequentially (12). As the proposed solution combines a global path planner followed by a local one, this time-dependent navigation stack applies for other situations where the robot must, for example, approach a person to interact with him instead of avoiding him or leading people to another room in the same environment.

This paper presents a new human-aware robot navigation framework based on time-dependent social mapping, where the path planning problem includes the use of the interaction spaces over time, restricting or penalizing the robot's path depending on the activities scheduled. This social information is added on top of the free-space graph which is later used for path planning and navigation. As the **main contribution**, this paper is the first one that includes and uses the concept of *chronemics* to plan socially accepted paths for robot navigation. For this purpose, this framework proposes a novel mathematical function for defining time-dependent social interaction spaces, which models how the passage of time affects the spaces where people interact with objects or other people. First, the navigation stack proposed in this paper uses a global path-planner algorithm based on a cost map. The result of this global planner is a socially accepted path that takes into account the spaces of interaction and time. The proposed framework uses a local planner that optimizes the trajectory. Finally, this work includes different experiments in simulated scenarios to validate the navigation stack. In these experiments, the robot plans socially accepted paths and navigates through the environment. In addition, we have conducted tests in real scenarios, a laboratory environment that simulates a caregiving center, to measure the proposal's performance in these situations.

This work is organized as follows: In section 2, a discussion of previous works related to robot navigation in human-environments is provided. Section 3 presents an overview of the proposed social navigation framework. Next, Section 4 describes the new model of time-dependent social interaction spaces. In Section 5, the socially-accepted path planning algorithm presented in this paper. Finally, Section 6 outlines the experimental results in both, real and simulated scenarios, and Section 7 summarizes the conclusions and future works.

2 Related works

The use of social robots for assistive environments is becoming more widespread in recent years. There are several works where socially assistive robots help older people in their daily tasks, for example, monitoring their behavior and activities, announcing events, accompanying them to therapy, or performing some activity as a virtual caregiver. For all this, the robot needs to perceive as much information as possible from the environment and the people in it. This task is not simple, and it is necessary for a complete system that connects the physical world (*i.e.*, the robot's perception system and the environment) with a cyber world that allows, from specific models and rules, to make decisions about the actions that the robot will perform. These Cyber-Physical Systems (CPS) extend the robot's capabilities to perform complex tasks. Fig 2 shows a diagram of a CPS system in an assistive environment, where the robot is just another sensor with the ability to move and interact with people and the environment itself. The basis of the CPS systems is an architecture capable of combining all the input data to make the right decisions.

In the specific problem of robot navigation in an environment with people, and in particular, the planning of socially accepted paths, a broad knowledge of the current state of the world facilitates the robot's ability to make appropriate decisions regarding the best trajectory during its navigation. In fact, the path planning problem in human environments is an essential task for the future generation of social robots. The way a robot navigates in real environments, such as an elderly care center, strongly affects our perception of the robot's intelligence (16). A path that explicitly takes

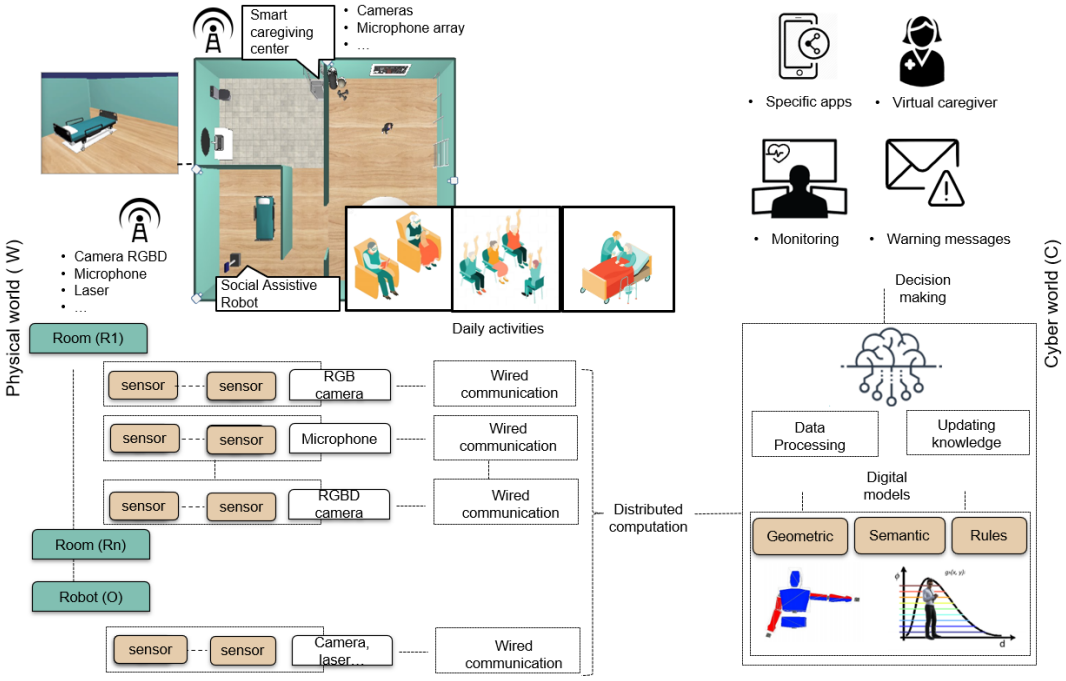


Fig. 2: General view of a Cyber-Physical System for caregiving center.

into account people and interactions human-human and human-objects should not only consider, for instance, minimizing the distance traveled to the target or the energy consumption, but also social rules (*e.g.*, keeping a comfortable distance to people or not disturbing them during interaction with the objects in the environment).

This type of human-awareness navigation is known in the literature as social navigation, a topic that started being extensively studied in the last decade. Since then, several navigation frameworks based on different theories and methods have been proposed. As mentioned before, most of them based on the concept of proxemics. In the recent study presented in (17), authors analyze the impact of robot motion on humans according to social hierarchy and the socio-physical context, concluding that the way in which navigates has an impact on the interaction that emerged between them. Several works, such as those presented in (18; 19), and more recently (20), describe the main approaches to the current state of the art. As a summary of these works, it is concluded that the main capabilities that a social robot should exhibit while navigating in an environment with humans are the ability to respect personal spaces for interaction and to avoid sudden and noisy movements that cause distraction. We have taken into account these two social skills in our approach.

Human-aware navigation frameworks face a significant challenge, combining deliberative and reactive behaviors during navigation: a social robot needs to be reactive to the dynamic environment (moving people) and plan a socially accepted path. The latter, path planning, is one of the classic problems of autonomous robot navigation. How the robot decides which is the best route from origin to destination has been extensively studied in the literature (the readers can see recent surveys in (21),(22)). The aim is to get a set of waypoints that the robot approaches one after the other. This path should optimize the robot's performance according to a global objective function or cost function, *e.g.*, the shortest path (or the quickest solution) without collisions. Most classic path planning approaches in static environments use graph-based search methods, with a graph representing states in the map. In this graph, a node is a point where the robot can stand without

collisions, and an edge between two nodes means that the robot can navigate between the nodes without causing a collision. The type of graph (*e.g.*, square grids, arbitrary lattices, or expanding random trees) influences the planning strategy. In our proposal, we use search algorithms on a grid of the environment, which, in general, is computationally efficient and provides optimal paths (18). Our proposal can be integrated with any navigation algorithm that uses a weighted graph, including Dijkstra, A*, or D*, for instance. Dijkstra's algorithm was conceived by E. W. Dijkstra in 1956 and published in 1959 (13) to find, in a weighted graph, the shortest path from an origin vertex to the rest of all vertices. The A* algorithm was published in 1968 by Peter Hart et al. (14). Its objective is to find a minimum cost path between two nodes of a weighted graph. To do so, it maintains a subgraph of paths originating from the starting node and expands these paths, trying to minimize a function cost (shortest distance traveled, shortest time, etc.). The Dynamic A* search algorithm (D*) was published in 1994 by Stentz (15). As well as A*, its objective is to find the lowest-cost path between two nodes of a weighted graph. Concerning the previous ones, the particularity of the D* algorithm is that the edge's weight can change during the problem-solving process, replanning the lowest-cost path online.

Furthermore, in dynamic scenarios, the robot's motion plan needs to be frequently updated to consider the changes in the environment. In these conditions, the path-planning problem is usually divided into two stages: global path planning and local motion planning. From this point on, social navigation algorithms extend the classic problem to environments with people by adding other parameters and constraints to the global path; that is, they model social conventions by using specific cost functions (18). Then, a local plan, updated at high frequency, is used to follow the global path and deal with previously unknown dynamic obstacles. The most widely used solutions for these short-term plans in the literature are the classic Dynamic Window Approach (DWA) (23), Potential field (24), Elastic bands (25), Reciprocal Velocity Obstacles (RVO) (26) or Social forces models (SFM) (27) approaches. In our proposal, we use an elastic band as a local path-planner, which main advantage is that the forces of attraction and repulsion are combined, and the path is always connected to the target. This structure is richer and facilitates replanning when there are blockages and the possible introduction of additional constraints on the shape of the path or the range of the robot control variables. In (28), the authors summarize the last path-planning strategies in these dynamics environments, distinguishing between classical hierarchical planners (global and local path-planners) and reinforcement learning-based ones. From these works, it is concluded that there is no single best or worst strategy, and for all of them, there is still room for improvement. In conclusion, human-aware path planning strategies are diverse, and the best strategy in aspects such as comfort, safety, or naturalness is challenging to define, which are sometimes related to people's psychological and sociological factors.

Some authors propose models of social rules by using cost functions (29; 30). A typical solution is to add social conventions or social constraints. For instance, Sisbot et al., (31) included social constraints associated with safety, comfort, visibility, and hidden zones into their cost model. In (29), the authors present a classical A* path planning method in conjunction with social conventions, like to pass a person on the right side of a corridor. In (30), the authors use potential fields and a proxemics model to define regions where the robot can navigate. Recently, some researchers differentiates between human beings and objects to generate human-friendly paths that maintain humans' safety and comfort during the robot's navigation (32). In (33), the authors propose a generative navigation algorithm in an adversarial training framework that learns to generate a robot's path that is both optimized for achieving a goal and for complying with latent social rules. A novel planning framework for social robot navigation is described in (34), where authors combine implicit (robot motion) and explicit (visual/audio/haptic feedback) communication, besides a predictive model of human navigation behavior. In all these works, the path planning algorithm adds restrictions to the routes without considering changes over time, which could be its most significant drawback.

Most works in the robotics literature address robot social navigation in interactive environments with people as a problem of *social mapping*, whose primary strategy is to define social interaction spaces in which robot navigation is forbidden or penalized (8). An usual strategy in social navigation is to map the space around a person at a higher cost than the free space. In this way, the use of this space by the path planner is penalized. In the extreme case, a forbidden zone for navigation can be defined as an obstacle, causing the planner to adjust its route not to cross it. In (8; 18; 19), authors define areas in the people's surroundings in which the robot's navigation is adapted by using the concept of proxemics. In the recent work presented in (35), authors use the Extended Social Force Model (ESFM) to describe the interactive force with human and environment. This last concept is not new, and other works use similar approaches (20). Other social navigation model is proposed in (36), where they uses context extraction from ontology. Similar works use the term affordance or activity spaces and prevent robots from navigating near them, creating regions where navigation is also forbidden or penalized (37; 9). Recently, in (38), the concept of interaction spaces, and their use to define social paths were introduced. However, all previous authors and works consider these social maps as static, and there is no dependence over time. As a result, these social maps are unrealistic and, in some situations, produce inconsistent robot behavior that might feel unnatural and confusing.

In this respect, this time-dependent social mapping is poorly studied in the robotics literature. The relationship between social conventions and time has been coined in other disciplines, such as *chronemics*. This theory started in the field of social science, and it studies as time directly influences human interactions (10). The concept of temporal planning has been introduced in some works to provide changes in the path that consider the people's movements. Most of the earlier work is limited to collision-free trajectory planning. Kollmitz et al. (39), is among the few works found in the state of the art of our field, the authors predict pedestrians' movements using time-layered occupancy grid maps. In the paper, the authors do not consider the time dependency as it will be described in our proposal, but only to estimate those spaces where people can be in the future according to their velocities and current poses. Recently, in (40), (41), the authors present and extend, respectively, SocioSense, a real-time algorithm for human-aware navigation that computes people's personality or time-varying behavior and dynamically models the proxemics. The authors include this information in their temporal planning algorithm; however, the concept applies only to people in motion, whereas in our paper, the more general concept applies to people, objects, and the specific context. Thus, the proposed algorithm differs in terms of performance because it avoids directly navigating through spaces where many people concur, and that would cause a disruptive effect. As this first planning is deliberative, the robot's path is optimal and minimizes sudden movements in the robot to adjust to people's movements. In the paper (42), the author defines a new general concept, the *behavioral map*, which comprises the monitoring of the human's presence in specific places during the time and label these spaces related to human action. The author does not use this definition for any specific scenario but points out that it can be used to plan a path that avoids people.

Consequently, our proposal has been inspired by some of the concepts presented, such as proxemics, chronemics or affordance spaces. As the main novelty, it is defined the time-dependent social mapping, taking into account the activities agenda of an elderly care center and how it influences in the social interaction spaces. The proposal uses the classical Dijkstra's algorithm, where the weights of the nodes are modified to take into account the social map of the environment and its dependence over time. Our approach is agnostic to any other planning strategies that uses a graph for representing states in the map (collisions) and graph-based search methods.

3 Overview of the social navigation framework

Robotic social navigation in care facilities, where the center's professionals schedule all activities, requires a reformulation of the classic social navigation algorithm, as well as the use of an evolved hardware and software architecture. This work uses a shared representation of the environment called Deep State Representation (DSR) and the CORTEX cognitive architecture, both described in (11). DSR is a multi-labeled graph that stores the environment's information: rooms, humans, objects, and the robot, among others. In this graph, nodes are the elements (*e.g.*, "room", "table"), and arcs are the relationship between them (*e.g.*, "in", "connected") (11). Software agents interact with this representation of the world to include new nodes (*e.g.*, a new person comes in a room, or a new object is detected) or update relationships (*e.g.*, two people start an interaction or the robot moves to another room). Fig. 3 illustrates a simplified example of the DSR graph for an elderly care center.

CORTEX cognitive architecture must be complemented by a perception system that monitors people and objects in the environment. The navigation framework presented in this article, although not strictly necessary, requires an understanding of what is happening in the care facility far beyond the robot's perception system. In other words, in this proposal, the smart-environment will solve the problem of this detection and tracking of people's position, the detection of changes in the objects' position, and the interactions between people and objects, using an RGBD cameras array strategically distributed throughout the caregiving center.

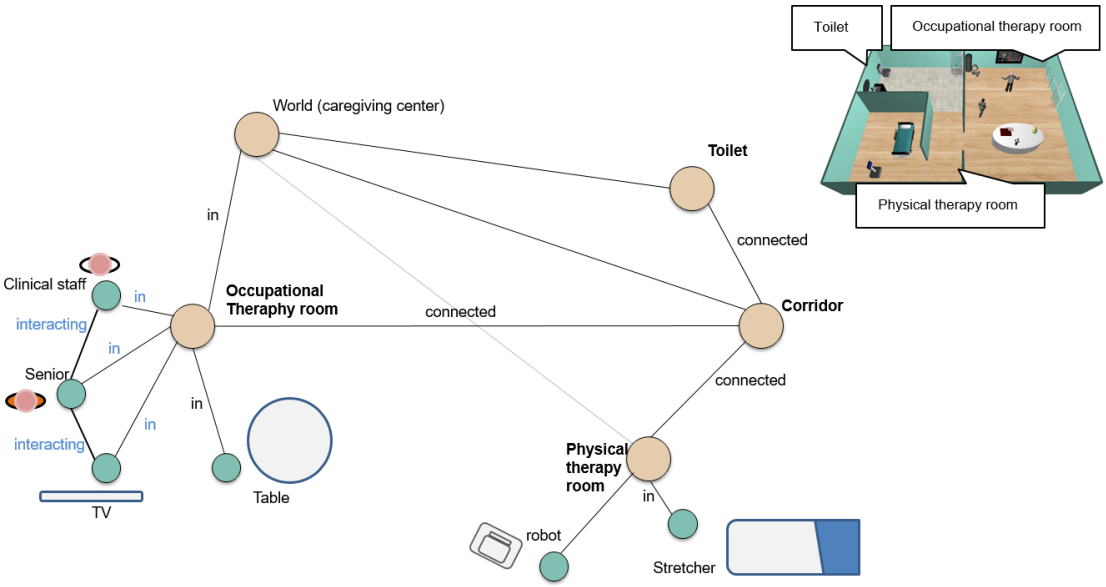


Fig. 3: An example of the Deep State Representation in an elderly care center.

The overview of the proposal is described in Fig. 4. The social path is solved using a classical Dijkstra algorithm that uses a free-space graph, $G(N, E)$. In this free-space graph, each node's weight varies according to the time-dependent social map generated by the cognitive architecture. The planned route is updated as it is traveled by using the elastic band path optimization algorithm described in (43), adapting it to unexpected events, such as dynamic obstacles. The time-dependent social map is built according to the information provided by the Deep State Representation, DSR, which has been previously updated from two different agents: the human observer agent and the

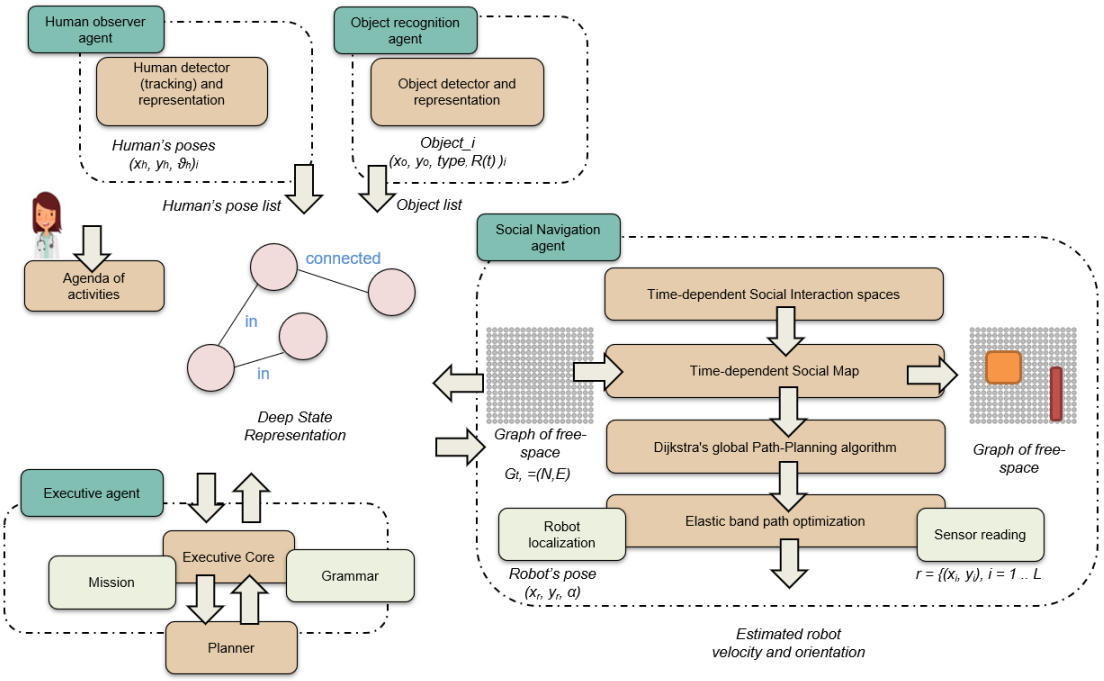


Fig. 4: Proposed system architecture.

object recognition agent. The first one, the human observer agent, is in charge of detecting and tracking people in the scene. The object recognition agent is responsible for detecting objects and monitoring their pose in the environment. Finally, the time dependence of this social map is provided, among others, by the center's professionals.

All agents in the architecture communicate with the DSR through the Executive agent. This agent is responsible for planning feasible plans to achieve the current mission, managing the changes made to the DSR by the agents as a result of their interaction with the world, and monitoring the execution of the plan. The Executive agent uses a visual planning domain definition language named Active Graph Grammar Language (AGGL), and an AGGL-specific planner (46) based on the Planning Domain Definition Language (PDDL). The agents involved collaborate by executing the actions of the stages of the plan. Each time a plan step is completed, a change in the model is included. The Executive agent uses this current state of the model, the domain, the target, and the previous plan to update the current plan accordingly. The following sections delve into the navigation framework devised in this article.

4 Time-dependent social interaction spaces

Building a coherent social map of the robot's surroundings is one of the main targets of the proposed navigation framework. In real-life scenarios, such as caregiving centers, people interact with each other and objects in the environment. The first step of the presented framework implies the definition of the social interaction spaces associated with people and objects and their dependence over time.

4.1 Social mapping: people in the environment

Considering $S \subset \mathbb{R}^2$ the space of the global map, let $H_N = \{h_1, h_2 ... h_N\}$ be a set of N humans detected by the human observer agent, where $h_i = (x, y, \theta)_i$ is the pose of the i-th human in the environment, being $(x_i \ y_i)^T \in S$ and $\theta_i \in [0, 2\pi)$[1]. To model the interaction space of each person h_i an asymmetric 2-D Gaussian curve $g_i(x, y)$ is used, as described in (37). The Asymmetric Gaussian function is composed of two halves of 2D Gaussian functions: an elliptical function in one direction, and a different ellipse in the opposite direction. This function associates the distance between a point $\mathbf{p} = (x \ y)^T \in S$ and the person's position with a real value $g_i \in [0, 1]$ as:

$$g_{h_i}(x, y) = exp^{-(\gamma_1(x-x_i)^2 + \gamma_2(x-x_i)(y-y_i) + \gamma_3(y-y_i)^2)} \tag{1}$$

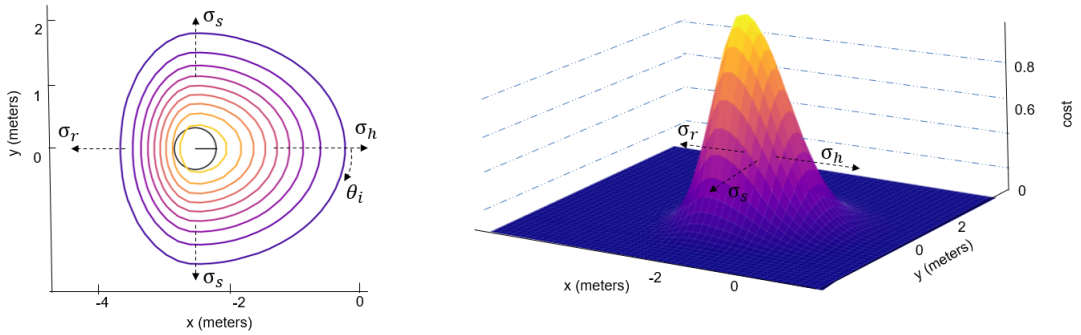
where the coefficients γ_1, γ_2 and γ_3 are associated to the rotation of the function θ_i. Let σ_s be the variance to the sides $(\theta_i \pm \pi/2)$, and let σ the variance along the person heading σ_h (θ_i), or the variance to the rear σ_r $(\theta_i \pm \pi)$, these coefficients γ_i are given by:

$$\gamma_1(\theta_i) = \frac{cos(\theta_i)^2}{2\sigma^2} + \frac{sin(\theta_i)^2}{2\sigma_s^2} \tag{2}$$

$$\gamma_2(\theta_i) = \frac{sin(2\theta_i)}{4\sigma^2} - \frac{sin(2\theta_i)}{4\sigma_s^2} \tag{3}$$

$$\gamma_3(\theta_i) = \frac{sin(\theta_i)^2}{2\sigma^2} + \frac{cos(\theta_i)^2}{2\sigma_s^2} \tag{4}$$

One aspect of the proxemics theory is the idea of interaction spaces are greater in front of people, and least behind. Besides, this personal space tends to have the same basic shape across cultures (29). Thus, the values of these variances are fixed to $\sigma_h = 2$, $\sigma_r = 1$ and $\sigma_s = 4/3$. Fig. 5 depicts various views of one such Asymmetric Gaussian cost function. The function shown is centered at (-2.5, 0), has a rotation of $\theta_i = 0$, and has these values of variances. The maximum cost is 1.0 at the center of the function.

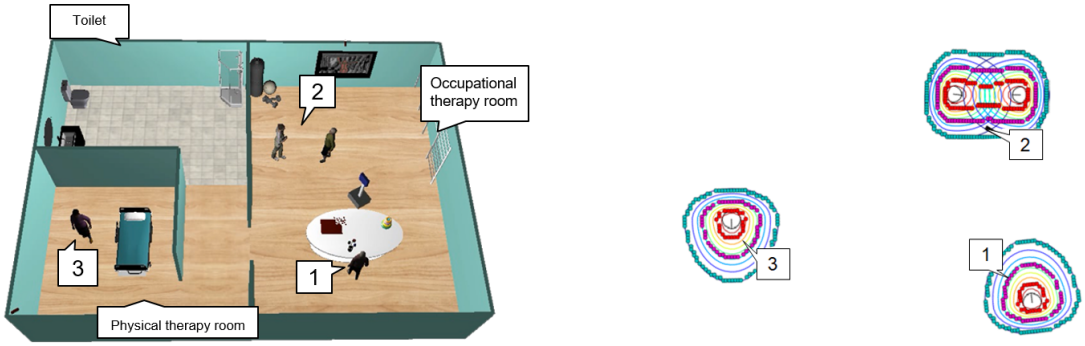


(a) (b)

Fig. 5: Two different views of an Asymmetric Gaussian function centered at (-2.5, 0), rotated by θ = 0, and having variances σ_h =2, σ_s = 4/3, and σ_r = 1.0; a) contour map; and b) surface map.

Fig. 6a shows a scene of a simulated caregiving center. There are two different rooms, a toilet and a corridor, and four people inside. Two of the people interact with each other, and the other two are alone in their respective rooms. Two people, labeled as 1 and 3 and their personal spaces modeled by asymmetric Gaussians, are drawn in Fig. 6b.

[1] The actual detection of people is out of the scope of the paper. In the experiments carried out it was performed by the CORTEX architecture and a RGBD camera network.



(a) (b)

Fig. 6: a) People in a simulated caregiving center; b) asymmetric Gaussian associated with person 1 and 3, and clustering of the two people labeled as 2 in Fig. 6a.

The algorithm clusters the humans detected in the environment according to their distances by performing a Gaussian Mixture (37). Let $g_{h_i}(\mathbf{h})$ be the personal space function for each individual i in the set of all H_N of all people in S. The global density space function $G_d(\mathbf{h})$ is defined as:

$$G_d(x, y) = \sum_{i \in H_N} g_{h_i}(x, y). \tag{5}$$

In order to group individuals into clusters, the method chooses the Ω_d and Ω_θ parameters as the smallest Euclidean distance and the smallest difference of angles between two people $\mathbf{h_i}(\mathbf{x}, \mathbf{y}, \theta)$, $\mathbf{h_j}(\mathbf{x}, \mathbf{y}, \theta) \in H_N$ such that those two are neighbors. These values are given by the insights of proxemics (the assessment of these parameters was described in (37)). If $\mathbf{h_i}(\mathbf{x}, \mathbf{y})$ and $\mathbf{h_j}(\mathbf{x}, \mathbf{y})$ are neighbors, then $\|\mathbf{h_i}(\mathbf{x}, \mathbf{y}), \mathbf{h_j}(\mathbf{x}, \mathbf{y})\| \leq \Omega_d$ and $\|\mathbf{h_i}(\theta), \mathbf{h_j}(\theta)\| \leq \Omega_\theta$, and the density contribution δ between them is:

$$\delta = g_{h_i}(\mathbf{h_j}). \tag{6}$$

Since $g_{h_i}(\mathbf{h_i}) = 1$ for each $\mathbf{h_i} \in P$, then if $\mathbf{h_i}$ has k neighbours then $G(\mathbf{h_i}) \geq 1 + k\delta$. Hence, the method can adjust a density threshold ϕ in order to group individuals who have at least k neighbours. ϕ is given by:

$$\phi = 1 + k\delta \tag{7}$$

and it can compare the value of the global function for each point in S and determine whether that point belongs to the personal space of a group of individuals. The set of such points is denoted by J and given by the expression:

$$J = \{\mathbf{h} \in S \mid G_d(\mathbf{p}) \geq \phi\} \tag{8}$$

Finally, the contours of these regions are defined by a set of k polygonal chain (*i.e.*, polyline) $L_k = \{l_1, ..., l_k\}$, where k is the number of regions detected by the algorithm. The curve l_i is described as $l_i = \{a_1, ..., a_m\}$, being $a_i = (x, y)_i$ the vertices of the curve, which are located in the contour of the region J_i.

According to (19), it is possible to classify the space around a person into four zones, depending on the degree of social interaction: public, social, personal, and intimate. Each human h_i present in the environment will have three associated spaces: the intimate space, delimited by the polyline ${L_k}^{intimate}$; the personal space, delimited by ${L_k}^{personal}$; and the social space, delimited by ${L_k}^{social}$, each of them being larger than the previous one, as it was introduced in (19). The public region

will be the remaining free space. These contours, which are created by choosing different values of the density threshold ϕ, can be seen in Fig. 6: in red is shown the intimate space, in purple the personal space, and in blue the social space.

However, these social spaces do not vary according to time. Consequently, our approach proposes adding a simplified version of the concept of chronemics in the social interaction spaces defined above. Interactions between people and people with objects follow socially accepted conventions (47). For example, interruptions are usually more annoying at the beginning or in the middle of the action than when it is nearing completion.

4.2 Social mapping: objects in the environment

In caregiving centers are common to perform physical or cognitive therapies where people - elderly or professional - interact with objects. Robots should be able to detect these situations before planning their path. As mentioned, the literature defines the concept of *Affordance spaces* to refers to areas where humans usually perform particular activities (9). These spaces are related to the way people interact with the object, and they are different for each type of object. These spaces are called *Activity spaces* when people are interacting with them, and they are, in general, fixed regions that have been mapped a priori as forbidden or free spaces for robot navigation. As before, these spaces do not vary according to time. In order to achieve greater flexibility in path planning and a higher degree of socially acceptable navigation, in our approach, this interaction space is time-dependent.

Therefore, let $O_M = \{o_1, ..., o_M\}$ be the set of M objects with which humans interact in the environment. Each object $o_k \in O_M$ stores as attributes its pose $p_{o_k} = (x, y, \theta)_k$, being $(x_i,\ y_i)^T \in S$ and $\theta_i \in [0, 2\pi)$, and its interaction space i_{o_k}. This interaction space is composed of the affordance space that describes the human-object interaction A_{o_k} and the time-dependence $R_{o_k}(t)$. Therefore:

$$o_k = (p_{o_k}, i_{o_k}) \tag{9}$$

where $i_{o_k} = (R_{o_k}(t), A_{o_k})$.

Different objects in the environments have different interaction spaces i_{o_k}. For instance, when using a table for therapies, a smaller area is needed compared to when watching television because this can be done from a farther distance. These areas are modeled by using the affordance spaces A_{o_k} for each object o_k. Besides, these interaction spaces include the chronemics and the time-dependence according to the $R_{o_k}(t)$ function. Thus, depending on the object's interaction space and the way that people interact with, i_{o_k} is modeled using one or another model. In this paper, these spaces have been modeled and classified as: i) isosceles trapezoidal shapes, a_t (*e.g.*, TV or poster); ii) rectangle shapes i_r (*e.g.*, tables, beds or stretcher); and iii) circular shapes i_c (*e.g.*, circular tables). a_t, a_r, and a_c define the occupied areas that describe the affordance spaces for trapezoidal, rectangular, and circular shapes, respectively. Fig. 7 illustrates the interaction space of each type of object.

- *Trapezoidal shapes*: Object like TVs are common in caregiving centers. Today, with the incorporation of therapies based on Serious Games (48), their presence is even more frequent. This type of objects are modeled as an isosceles trapezoid with height t'_h and widths (t'_{w1}, t'_{w2}), as described in (37).

$$a_t = t'_h \cdot \frac{t'_{w1} + t'_{w2}}{2} \tag{10}$$

- *Rectangle shapes*: Objects like tables, beds, or stretchers are rectangular objects typically used in caregiving centers. These objects are modeled as a rectangle with height r'_h and width r'_w.

$$a_r = r'_h \cdot r'_w \tag{11}$$

– *Circular shapes*: Objects like circular tables are also common in caregiving centers. These objects are modeled as a circle with center in p_{o_k} and radius c_r'.

$$a_c = \pi \cdot {c_r'}^2 \tag{12}$$

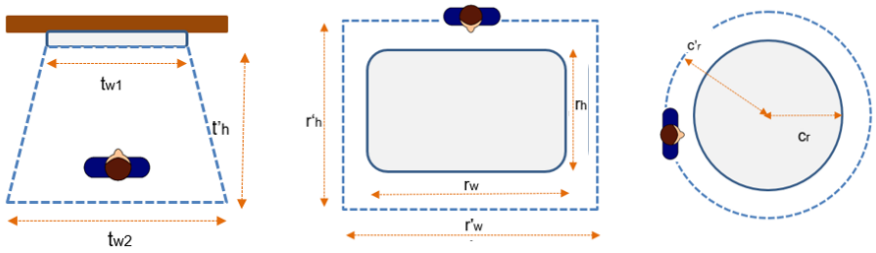


Fig. 7: Object interaction space is modeled by: isosceles trapezoid (left); rectangle (middle); and circular shapes (right)

In this paper, $R_{o_k}(t)$ is a mathematical function which ranges from $R_{min} \leq R_{o_k}(t) \leq R_{max}$, where R_{min} and R_{max} mean that the object does not have any activity scheduled at that time t or it has an activity scheduled at this time, respectively. When the start time of the activity is approaching, the value of $R_{o_k}(t)$ increases. On the contrary, when the activity finishes, the value of $R_{o_k}(t)$ decreases. In summary, $R_{o_k}(t)$ could be understood as a countdown for the remaining time until the start of the activity and a count for the elapsed time after the end of the activity. Fig. 8 shows an example of $R_{o_k}(t)$ for a generic object.

Finally, a polyline for each object $L^t_{o_k}$ is defined. The set $L^t_o = \{L^t_{o_1}, ..., L^t_{o_M}\}$ describes the set of polylines used by the navigation framework at an instant time t.

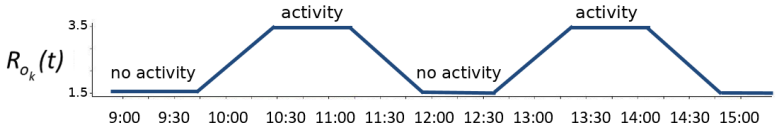


Fig. 8: $R_{(t)}$ function of time-dependent social interaction space for a generic object.

5 Socially-acceptable navigation framework

This section describes the social navigation framework proposed. First, the social map is integrated into a social path planner, which is executed globally at the beginning of the trajectory, and every time a significant change in the environment is detected. In this global path-planner, the robot's environment is represented by a uniform graph composed of obstacle-free nodes with a fixed finite traversal cost, and non-free nodes, which have an infinite one. The proposal modifies the costs according to the social map, and the optimal path uses the Dijkstra's algorithm. Finally, a local planner based on elastic bands adjusts the path according to the readings of the robot's sensors.

5.1 Graph-based grid mapping

Space is represented by a graph $G(N, E)$ of n nodes, regularly distributed in the environment. Each node n_i has two parameters: availability, d_n, and cost, c_n. The availability of a node is a Boolean variable whose value is 1 if the space is free, 0 otherwise. c_i, indicates the traversal cost of a node. High values of c_i indicate that the robot should avoid this path, and low values of c_i indicate that the robot should use this path. Initially, all nodes have the same cost 1. Fig. 9a shows an original free-space graph in which all nodes have the same cost and availability.

The classical Dijkstra algorithm is employed for determining the shortest path between an initial position and a target to which the robot must travel. The algorithm calculates the cost from the node origin to the target node, taking into account the traversed nodes' cost. The cost of a path is the sum of the cost of the nodes that compose it and the path with the lowest cost will be selected.

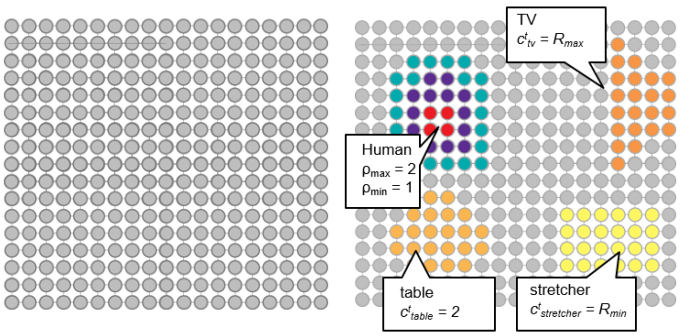


Fig. 9: Graph-based grid mapping: a) initial free-space graph; and b) final free-space graph, after including the social interaction spaces.

5.2 Social graph-based grid mapping

The free space graph is modified to include social interaction spaces. Firstly, those associated with the interaction between one person and another -or groups of people-. Secondly, those associated with the interaction between people and objects.

5.2.1 Personal space mapping

Being A the matrix formed by the availability of each node of the free space graph and C the matrix formed by the costs and considering the set of polygonal curves defined bellow, ${L_k}^{intimate}$, ${L_k}^{personal}$, and ${L_k}^{social}$, this paper presents the modification of the cost and availability of the nodes of the graph according to these interaction spaces.

Firstly, considering only the intimate space around the person h_i, for each polyline $l_i^{intimate}$ is defined a polygon $P_i^{intimate}$ formed by the points of the polyline. The availability d_{h_i} of all the nodes $N_i \in N$ contained in the space formed by $P_i^{intimate}$ is set to occupied, $d_{h_i} = occupied$. This means that the robot will not be able to invade this space, as it would disturb the person. For personal and social spaces, the nodes' availability is not modified, but its cost will be changed.

Secondly, considering the personal space around the human h_i, for each polyline $l_i^{personal}$ a polygon $P_i^{personal}$ has been defined. The cost c_{h_i} of all the nodes $n_i \in N$, contained in the space formed by $P_i^{personal}$ will be modified and set to $c_{h_i} = 4.0$. In the same manner, for the social space, a polygon P_i^{social} is defined for each polyline $l_i^{personal}$. All the nodes $N_i \in N$ contained in the space formed by P_i^{social} will have cost $c_{h_i} = 2.0$. The public space will be the rest of the graph whose costs remain unchanged. Fig. 9b show the final free-space graph, where the costs of nodes are modified according to the social spaces of interaction.

Intimate areas are forbidden for navigation. Personal and social spaces are available, but their costs are higher, being personal spaces more expensive than social spaces, starting from the free-space, the cost doubles when the robot needs to use a more personal space in its navigation. In this way, when the robot plans the shortest path, it will move away, reasonably, from the person, exhibiting better human-aware navigation. One of the main advantages of not considering the social and personal spaces occupied is that, if the robot does not have enough space for navigating, for example in a corridor, it will not be blocked, but will navigate in the social space, although the cost of the path will be higher. If the robot has no other alternative, it will cross the personal space, but it will never cross the intimate space.

5.2.2 Object space mapping

This same technique has been used for the objects. Firstly, for each object o_k is defined a polygon P_{o_k}, that represents the availability, d_{o_k}, of all the nodes $n_i \in A_{o_k}$ contained in the space formed by the polygon P_{o_k}. The availability of the nodes of each object, d_{o_k} in the matrix A_{o_k}, is set to *occupied*, $d_{o_k} = occupied$, while availability of the rest of nodes is not modified.

Secondly, let ${L^t}_o = \{{A^t}_{o_1}, ..., {A^t}_{o_M}\}$ be the set of polylines that describe the social interaction space for each object. For each ${A^t}_{o_k}$ is defined a polygon ${P^t}_{o_k}$ formed by the points of the polyline, which maps these points with the costs of the nodes in $G(N, E)$. Accordingly, the nodes' cost in the matrix C are set to $c_{o_k}^t = R_{(t)}$ in the free-space graph. These values are associated with the scheduled activities for each object.

Fig. 8 shows the evolution in the cost of going through a node in function of time. The value starts at $c_{o_k}^t = 1.5$, which means that there is no activity scheduled for that object or that remaining more than 45 minutes to its start. Likewise, the cost ends at $c_{o_k}^t = 3.5$, which means that an activity is being performed at that time. Finally, after the end of the activity, the cost decreases in the same way that it increased until reaching $c_{o_k}^t = 1.5$. $R_{(t)}$ function can be easily adjusted according to the caregiving center's specific needs.

5.3 Socially acceptable robot navigation

Once the time-dependent social map of the world has been defined, the next step is the planning of the socially accepted path and the final navigation to the target. The framework defines this step as a three-level hierarchy. These three levels are depicted in Fig. 10, and are defined as:

- Path planning algorithm: The time-dependent social map is used to generate global trajectories to specified targets.
- Elastic bands: The planned route is deformed in real time to handle local changes in the environment detected by range sensors. An elastic band is defined as a deformable collision-free path (25). The basic algorithm defines imaginary forces at each point along the robot's path. These imaginary forces are divided into an internal contraction force (f_c) and an external repulsion force (f_r). The first simulates the tension in a stretched elastic band and removes any slack in the path. The second counteracts the contraction force and gives the robot clearance around the obstacles. The two forces deform the elastic band until reaching the equilibrium. To

illustrate the basic behavior of elastic bands, consider a robot and three consecutive points on the planned path p_{i-1}, p_i, p_{i+1} (see Fig. 11). These three points define the contraction force. The readings of the range sensors define the repulsive force. The combination of the two forces modifies the final position of the path point p_i^{t+1}.

– Control: A feedback control law is used to move the differential robot along the elastic band.

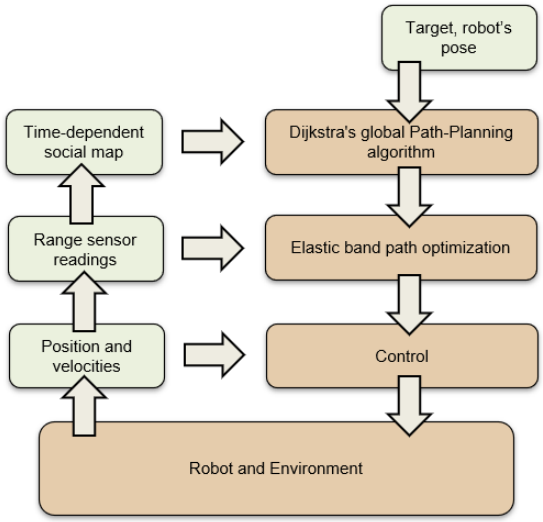


Fig. 10: A three level hierarchy for the proposed framework.

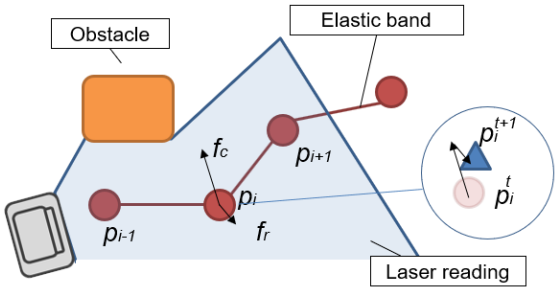


Fig. 11: The basic elastic band algorithm defines imaginary contraction and repulsion forces, f_c, f_r, respectively. These forces modify the band until equilibrium is reached.

The core data structure of the path is a list of 2D points or bubbles that behave as an elastic band under various internal and external forces and processes (25). The path is defined as the ordered set, $P = p_i : p \in \mathbb{R}^2 x \mathbb{N}, i \in 0..N$, with two real coordinates in a global reference system maintained by the robot, and an integer one for the bubble's radius. Each radius is computed as the minimum distance to the surrounding objects in the environment, using range sensors. The function $\rho(p)$ that computes this radius is defined as $\mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \{\mathbb{R}^+ \cup 0\}$ and is implemented as

a search iterating over the laser array and the list of visible points. The set of forces and processes that affect the path are:

- The Path planning algorithm creates a new path given the robot's position, the target, and the time-dependent social map. The creation of the path initiates the internal dynamics that will eventually take the robot to the target.
- Once an initial path is available, the size of the set P- is continously adapted by adding and deleting elements. The goal is to keep them equally spaced along the band's length even it is streched or shrinked. If two points are separated more than half the length of the robot, a new one is inserted. Conversely, if two points are closer than half of the robot, they are combined into a single point on the trajectory.
- The next process exposes the band's elements to an internal force, f_c, that opposes to the local curvature. The effect of this force is to tighten the band, making it as straight as possible. The force is computed from nearby points as:

$$f_c = k_c\left(\frac{p_{i-1} - p_i}{\|p_{i-1} - p_i\|} + \frac{p_{i+1} - p_i}{\|p_{i+1} - p_i\|}\right) \tag{13}$$

  where p_i is the position of step i in the path. The force only vanishes if the three points are aligned. These internal forces are illustrated in green color in Fig. 12.
- The last process exposes the band to a repulsion force, f_r, which pushes the robot's trajectory from the obstacles. The value of this f_r depends on $D(x, y)$, that is defined as the minimum distance from the position of point p_i to the obstacles in the environment. $D(x, y)$ is measured by the robots' range sensors. For each point in the path, p_i, the direction of maximum variation of $D(x, y)$ with respect to points coordinates (x, y) of the obstacles is computed with a discrete Jacobian:

$$\frac{\partial D}{\partial p} = \frac{1}{2\delta}\begin{bmatrix} D(p - \delta x) - D(p + \delta x) \\ D(p - \delta y) - D(p + \delta y) \end{bmatrix} \tag{14}$$

  where D is the minimum distance function defined above, p is the point in the path and x and y are the point's coordinates. $\delta x, y$ are discrete displacements in the point's position. The Jacobian is multiplied by the difference between a maximum distance threshold D_0 and the current value of $D(x, y)$:

$$f_r = \begin{Bmatrix} k_r(D_0 - D)\frac{\partial D}{\partial p} & p < D_0 \\ 0 & p \geq D_0 \end{Bmatrix} \tag{15}$$

  where k_r is a global repulsion gain and represents the maximum distance up to which the force is applied.
  The effect of this force is to adapt the original path to the state of the real world, correcting planning errors caused by an imprecise world model, by a miss-localization of the robot or to the appearing of unforeseen obstacles like dynamic objects. The force acts on the band's points by repealing them with a magnitude inversely proportional to the distance that separates them. This action effectively relocates the path at a safe distance from the obstacles, increasing the robot's clearance, and providing the reactive component necessary for real time control.
  Finally, each point, p_i , in the path is modified according to the sum of the repulsion force, f_r, and the contraction force, f_c.

$$p_i^{t+1} = p_i^t + f_r + f_c \tag{16}$$

  These repulsion forces are illustrated in thin blue arrows in Fig. 12. Very quickly, the whole path evolves towards an equilibrium point where all forces cancel out (25; 43).

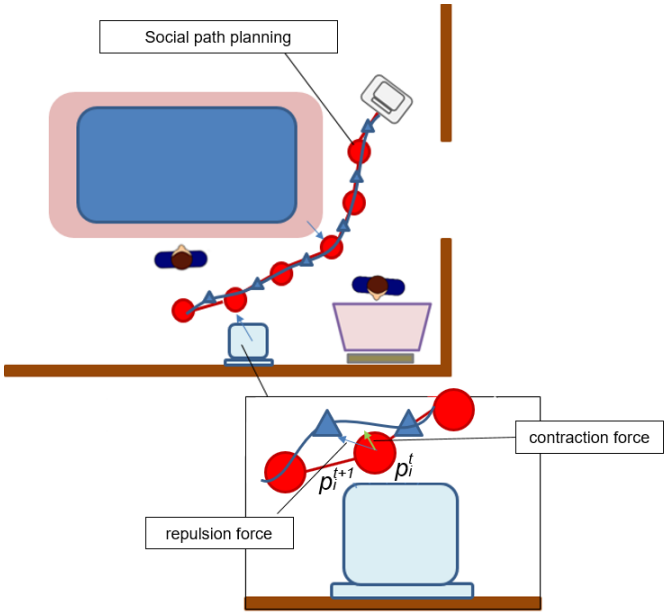


Fig. 12: The final social path is shown as the continuous blue line (△). Besides, steps provided by path planner (red circles), contraction forces (green arrow), and repulsive forces (blue arrow) are drawn.

5.4 Computation of optimal force gains

The elastic band algorithm provides smooth integration between global path planning and local path tracking. However, there are two free parameters that have to be manually tuned in order to achieve a good real-time response. These parameters are the gains applied to the forces that act on the band:

- k_c: the global constraint gain, that is related with the contraction force (see Eq. 13).
- k_f: the global repulsion gain, that is related with the repulsion force (Eq. 15).

An incorrect choice of these values and the path will nor adapt quickly enough to new obstacles, or it will become unstable under certain configurations of equally proximal obstacles. One way to find good values for a specific environment is to define a function over the robot's trajectories that provides higher scalar values for those well executed. Following this idea, a cost function is defined over the space of trajectories T as the linear combination of four penalty functions:

$$G = aH(t_i) + bD(t_i) + cQ(t_i) + dP(t_i) \tag{17}$$

where the coefficients a, b, c, d weigh the relative contribution of each term. H is sum of heading changes along the path,

$$H = \int_0^T d\theta dt \tag{18}$$

D is the total distance travelled by the robot,

$$D = \int_0^T \sqrt{(\frac{dx}{dt}) + (\frac{dy}{dt})} dt \tag{19}$$

Q is the total time used to complete the path, and P is the mean distance between the robot and people nearby along the path.

$$H = \frac{1}{T}\sum_{i=0}^{T} d^*(p_i, P), \quad with\, d_i^* = min_k(d(p_i, P_k^i)) \tag{20}$$

where $\{P^i\}$ is the set of people visible form position p_i

To minimize G over a sample of trajectories we use combination of direct search procedure and stochastic gradient procedure. One of the gains is fixed and the other one is sampled in a predefined range. For each value of the gain, the robot is sent to 20 different places and the resulting path is recorded. The functions H, D, Q and P are computed for each path t_i and combined into G_i. The path t^* that minimizes G is found along with its gain value. In the next iteration, this gain is set to the value corresponding to the minimum and the other gain is sampled along with a new set of paths. The procedure is repeated until no further improvement is found. Fig. 13 shows the different robot's paths, maintaining a fixed value of gain and varying the other. The left image shows different trajectories varying the k_c value and fixing k_f). The conclusion drawn from the experiments is that the optimized values for k_c and k_f are 120 and 60, respectively.

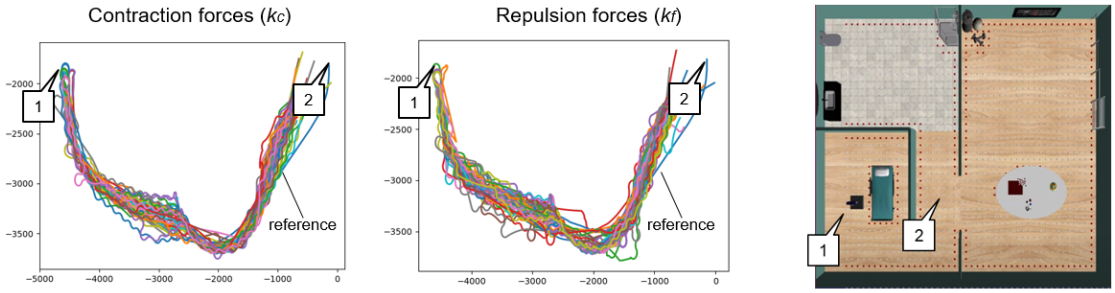


Fig. 13: left: robot's trajectories varying the k_c values during different tests; center: robot's paths where k_f values changes and the rest of parameters are constant; and right: simulated environment, where the initial and end position in the trajectory are labeled.

5.5 Planning of robot's actions during navigation

In the proposed framework, planning is performed with the symbolic information in the Deep State Representation (DSR), using the nodes of the representation as symbols and the edges of the graph as predicates (44). Fig. 14 illustrates an example of the shared representation associated to Fig. 6(a). As shown in Fig. 14, CORTEX uses a lot of different types of symbols and edges. However, only a set of symbols are used in the planning domain of the proposed framework: *human*, *robot*, *objects* and *room*. Similarly, the set of edges are limited in the planning domain.

This paper is focused on those cases where only a *robot* is located in the model, but where several people, objects, and rooms are possible. The planning domain in the navigation framework defines all the rules needed to be socially accepted, among others: to navigate without disturbing people, approach a person or group of people, get their attention, and initiate interaction, for instance. The planning rules are described through AGGL (44), and thus, they are defined as pattern pairs, in the same way as string grammar rules: each rule states that the pattern on the left-hand side can be replaced with the pattern on the right-hand side. For example, the *changeRoom* action in

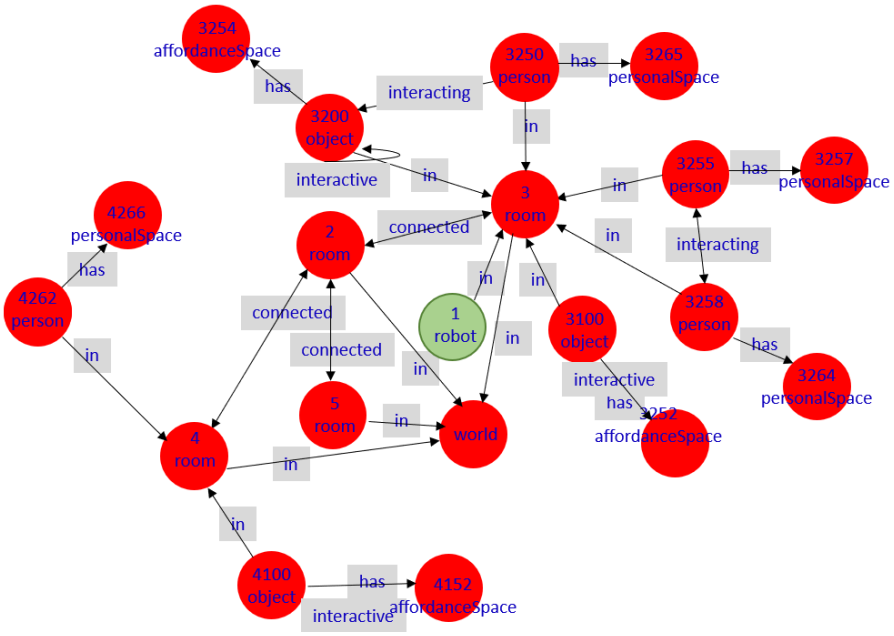


Fig. 14: Deep State Representation of the world shown in Fig. 6(a).

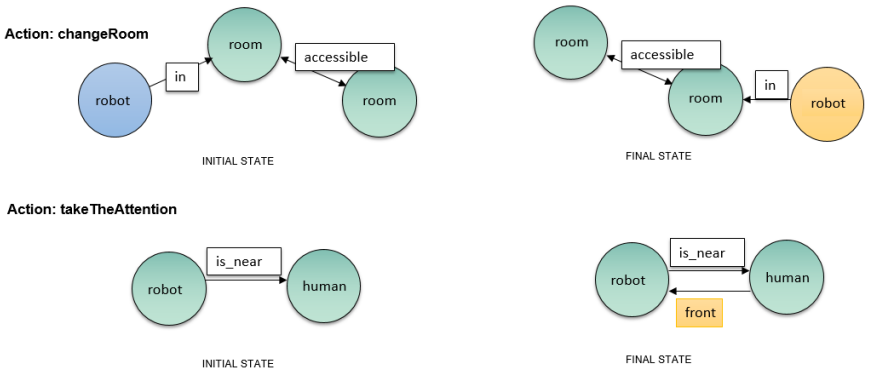


Fig. 15: Examples of rules defined in the navigation domain: *changeRoom*, where the robot navigate to other room in the environment, and *takeTheAttention*, where the robot navigates until being facing people.

Fig. 15 shows to the robot *in* a room in the left-hand side (initial state) and, on the right-hand side, after applying the rule, the robot should be *in* another room. In Fig. 15, the blue color indicates the nodes and edges on the left-hand side, in yellow, the elements that will be on the right-hand side, and in green the elements that do not change in the rule. Any changes in the care center's schedule are communicated through the DSR. Thus, the information in the other agents is updated in real-time, including the new actions described in the grammar. Any changes in the care center's schedule are communicated through the DSR. Thus, the information in the other agents is updated in real-time, including the new actions described in the grammar. This architecture facilitates the planning of specific actions in the robot. For example, if the robot wants to approach a person, such as the works described in (45) it generates a set of simple actions that the robot executes

sequentially: select the target person, estimate his or her pose, plan a trajectory that ends facing the person, and initiate the movement. The proposed navigation framework is integrated with this plan thanks to the DSR information (12).

6 Experimental Results

6.1 Validation of the Social navigation framework

To carry out the validation of the social navigation framework proposed in this article, simulated environments are used to ensure equal conditions in the study, thus isolating possible problems that may arise in real scenarios, such accuracy errors in locating objects and persons, and therefore, are exogenous to the test itself. The proposed time-based navigation framework applies to any semi-organized social environment whenever there is a schedule of activities. The simulated environment consists of a caregiving center. The evaluation requires the participation of the center's staff, who must schedule the activities in the center's agenda. The results are compared with a navigation algorithm without social behavior, evaluating metrics especially designed to validate the acceptance of the path performed by the robot.

For the reader's convenience, Fig. 16 shows the representation described in this paper for building time-dependent social interaction spaces. In Fig. 16a, a physical therapy room with three objects is shown (a TV, a circular table, and a stretcher). The center's professionals have set the activities schedule. In Fig. 16b shows a situation where there are no activities scheduled for any of the objects in the room. For each object, the shape of its social interaction space is observed. The yellow color serves to visually indicate that the cost of the node is the minimum. Fig. 16c and Fig. 16d show two other situations and how the time-dependent social interaction spaces evolve. As appreciated, the nodes' costs are represented by different colors for each object according to the activities agenda. In red to indicate that an activity is currently being executed. In orange to indicate that an activity has just been completed.

The social navigation framework has been developed in C++. The experiments have been performed in a PC with an Intel Core i5 processor with 4Gb of DDR3 RAM and Ubuntu GNU/Linux 18.10. Two different simulated scenarios have been designed. Both are composed of different elements, such as corridors, toilets, and physical therapy and occupational therapy rooms. There are professionals and older adults who carry out programmed activities. For people and objects' tracking, a network of RGBD cameras with an almost complete visual field of the scenarios must be deployed. Fig. 17 shows the two scenarios used in the simulations, as well as the RGBD cameras location. Fig. 18 shows images acquired by using the camera network deployed in the first simulated scenario at different instants of time. As the figure shows, there is minimal overlap between cameras, which is needed to monitor people and the robot during the activities. It is interesting to note that the simulation only refers to the scenario, everything else, image capture, path planning, and path optimization, laser data processing, etc. is done by the CORTEX architecture as if it were a real scenario.

The first experiment is shown in Fig. 19. This scenario (Fig. 17a) is composed of three rooms, objects, and people, where the activities schedules are modified. The first experiment consists of two tests. Firstly, the scheduled center activities are: (i) a therapy scheduled on the table, with an older person ($senior_1$) performing the therapy; (ii) no other therapy has been scheduled. Secondly, the situation changes slightly. In this case, there are therapies programmed on the table ($senior_1$) and on the TV ($senior_2$). In Fig. 19a and Fig. 19b the social paths executed by the robot for each test are drawn. These paths avoid crossing close to the people in the room, getting as far away from them as possible, always minimizing the distances traveled, and considering the time-dependent social interaction spaces.

In order to assess the validity of the proposed navigation approach, the methodology has been evaluated accordingly to the following metrics: (i) average minimum distance to a human during

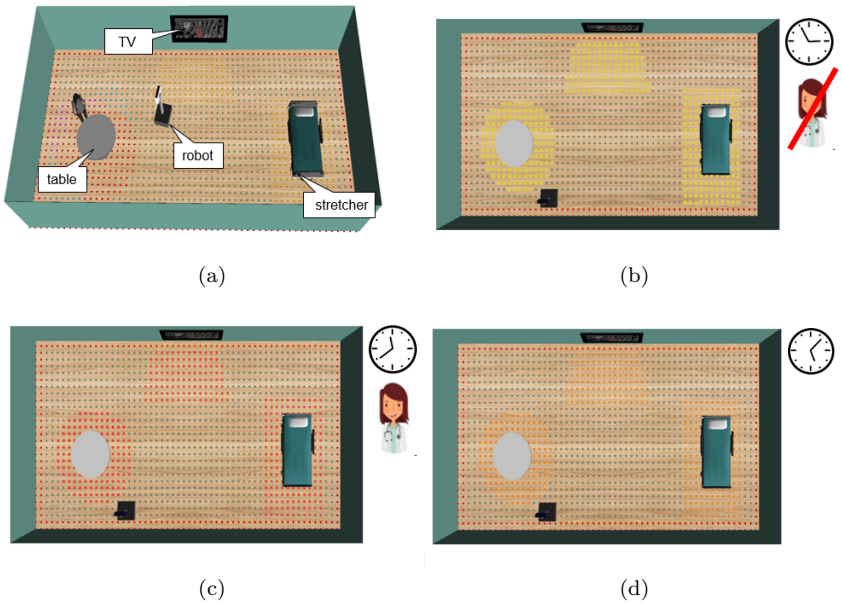


Fig. 16: First test: a) simulated physical therapy room with three objects inside. Time-dependent social interaction spaces and its costs in the graph: b) yellow, no activities scheduled; c) red, activities are being performed; and d) orange, activities have just finished.

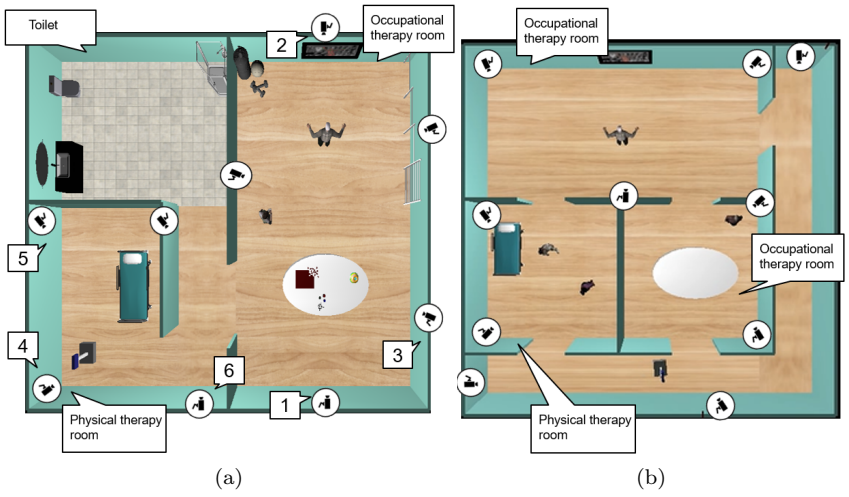


Fig. 17: Two different simulated scenarios. Both of them represent the typical room layout of real caregiving centers.

navigation, d_{min}; (ii) distance traveled, d_t; (iii) navigation time, τ; (iv) cumulative heading changes, CHC; (v) personal space intrusions, Ψ; and (vi) objects' spaces intrusions. These metrics have already been established by the scientific community (see (49; 50)).

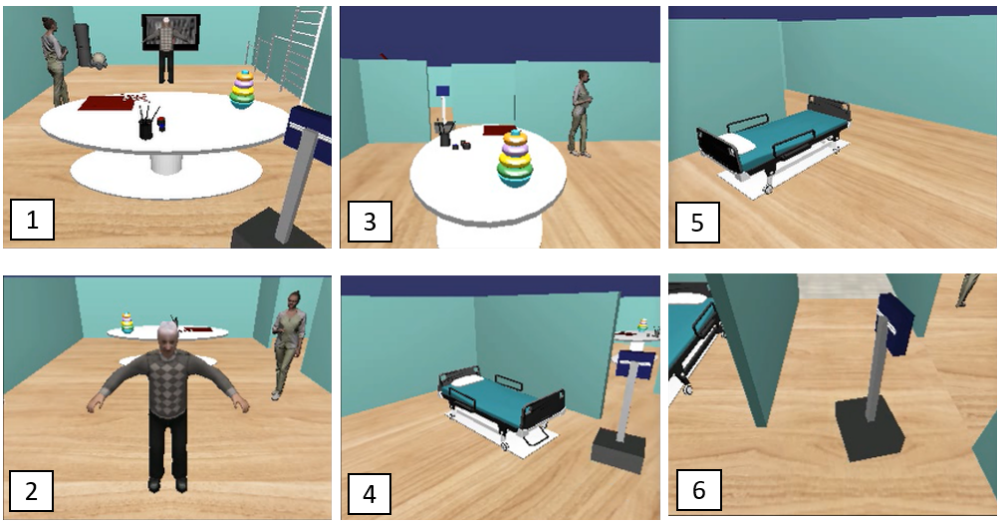


Fig. 18: An example of six RGB images acquired by the caregiving center's sensor network in different moments. The cameras are tagged to locate them in the first scenario used for the experiments. (See Fig. 17a)

The experiment results are shown in Table 1, where metrics for the performed path by a classical Dijkstra's algorithm without social behavior is also detailed. In Table 1, the performed paths by the proposed social navigation approach for both tests are identified as Path 1 and Path 2, respectively. First of all, as it is evident, the robot's performed path without social behavior travels a shorter distance in a shorter time. The time employee by the robot to reach its target increases compared to a classic path-planner without social behavior, but in return, it does not disturb people while they are performing their therapy.

However, and as Table 1 shown, for the non-social behavior algorithm, the distance to $d_{min}^{senior_1}$ (the senior that is opposite the TV) is minimal, which can bother the caregiving center's users. This same situation can also be observed with the value of $\Psi(Intimate)$, which indicates that the robot invades this usually forbidden space. In Path 1, the robot travels a shorter distance in less time, maintaining its social behavior without invading personal spaces, as shown in the data in the corresponding column. Also, in Path 1, the robot uses the TV's time-dependent social interaction spaces in its trajectory since it is not in use. In the Path 2, the robot also has a socially accepted behavior, adapting its trajectory to the center's activities, avoiding crossing the space associated with the TV. Consequently, it is guaranteed that the robot does not disturb during the therapies[2].

The second scenario was presented in Fig 17b. In this second experiment, the robot must navigate from its initial position to the occupational therapy room, where a user is performing an activity. During the robot's navigation, the activities programmed in other objects in the environment change according to the agenda, and the robot must re-planning its social path. Fig. 20a shows the robot's path at the time instant labeled as 1 in Fig. 20c. As shown in Fig. 20a, the planned path crosses near the table and the senior. At the instant of time labeled as 2, the table's time-dependent social interaction space changes, increasing the cost of the first planned path. Consequently, the robot creates a better social path, and this trajectory is shown in Fig. 20b. Table 2 describes the main results for this experiment, comparing the path finally carried out (Path2) with a classical navigation approach. Also, the results when the robot crosses the occupational therapy room during the therapy are shown (Path 1). From the results of this tables, it can be concluded that the

[2] First experiment video: `https://youtu.be/j-Lw5taqqDc`

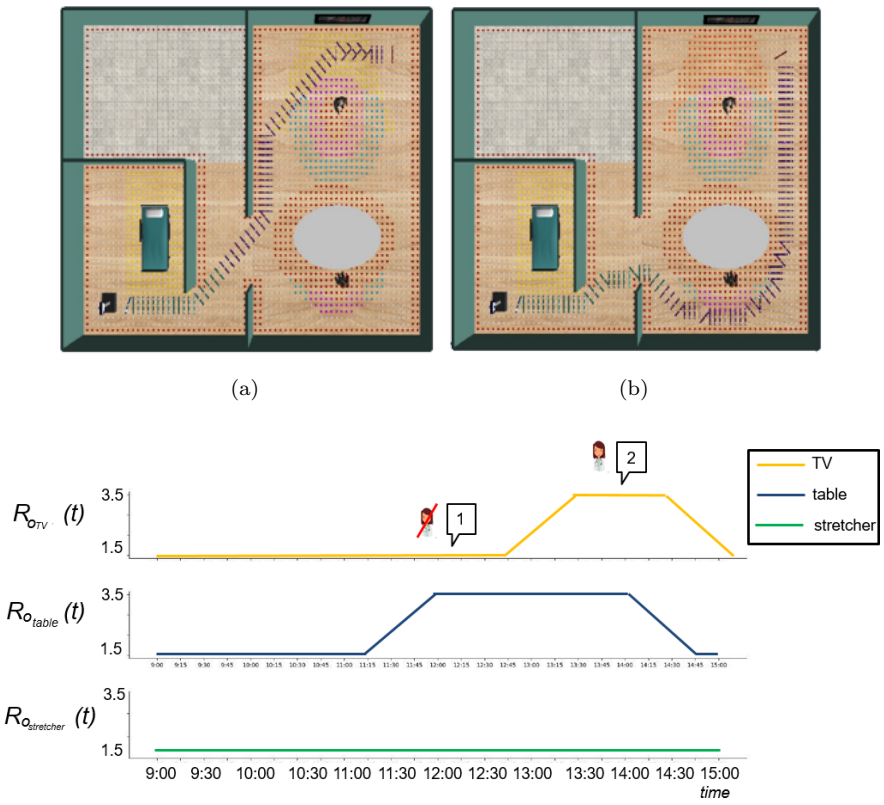


Fig. 19: First experiment. Scenario Fig. 17a: a) path planned at the time labeled as 1. There is an activity on the table; b) path planned at the time labeled as 2. There are two activities, one on the table and another on the TV; and c) center's activities agenda.

Table 1: Navigation results for the experiment shown in Fig. 19

Parameter	**Path 1** Value	**Path 2** Value	**Non social behavior** Value
d_t *(m)*	14.03	17.40	12.81
τ *(s)*	101.79	124.24	97.21
CHC	0.47	1.75	2.12
$d_{min}^{senior_1}$ *(m)*	1.39	1.89	0.51
$d_{min}^{senior_2}$ *(m)*	2.14	1.13	2.11
Ψ *(Intimate) (%)*	0.0	0.0	4.21
Ψ *(Personal)(%)*	0.0	11.0	0.0
Ψ *(Social)(%)*	0.0	8	12.44
Ψ *(Public)(%)*	100.0	81.0	83.35
Ψ *(Objects) (%)*	34.98	1.64	4.62

navigation framework described in this article improves the expected navigation results according to the exposed metrics[3].

[3] Second experiment video: `https://youtu.be/v6rG8uWhwKw`

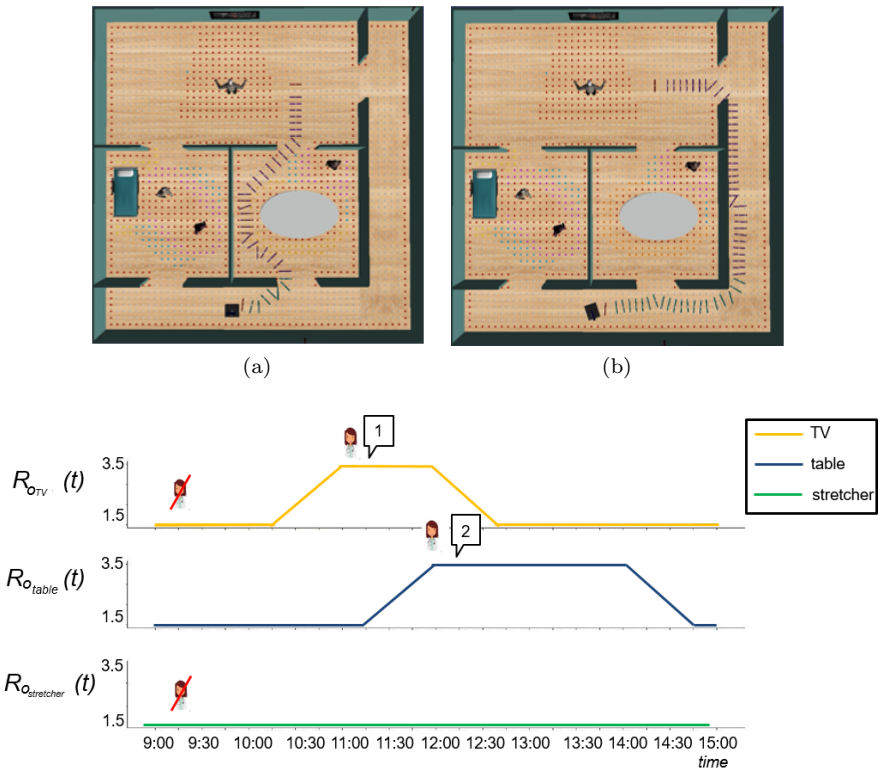


Fig. 20: Second experiment. Scenario Fig. 17b: a) path planned at the time labeled as 1; b) path planned at the time labeled as 2; and c) center's activities agenda.

Table 2: Navigation results for the experiment shown in Fig. 20

Parameter	**Path 1** Value	**Path 2** Value	**Non social behavior** Value
d_t (m)	11.30	16.28	10.92
τ (s)	52.5	63.1	51.1
CHC	9.28	3.5	12.12
$d_{min}^{therapist}$ (m)	3.16	5.17	3.11
$d_{min}^{senior_1}$ (m)	2.70	2.88	2.65
$d_{min}^{senior_2}$ (m)	1.82	3.41	1.42
$d_{min}^{senior_3}$ (m)	1.58	2.10	1.65
Ψ $(Intimate)$ $(\%)$	0.0	0.0	0.0
Ψ $(Personal)(\%)$	0.0	0.0	0.0
Ψ $(Social)(\%)$	16.47	0.0	18.03
Ψ $(Public)(\%)$	83.53	100.0	81.97
Ψ $Objects$ $(\%)$	69.85	0	73.22

Question	avg. (σ)
The robot navigates in a similar way to the human	3.86 (0.57)
The robot correctly approaches its target without interfering with the therapy	4.68 (0.48)
Did you feel uncomfortable during the robot's navigation?	4.57 (0.52)
The robot shows socially accepted behavior	4.37 (0.52)

Table 3: 10 participants used a Liker scale-based questionnaire to evaluate the robot's navigation during the real tests.

6.2 Human-aware navigation in real scenarios

For testing in real scenarios, we have used the semi-humanoid robot Viriato. It is an omnidirectional robot, approximately 1.6 m tall, and it has several onboard cameras for navigation (SLAM) and interaction with objects and people. It also has a laser sensor for object detection during motion. Fig. 21 shows the laboratory environment used for the tests. The assistance environment consists of a 65 m2 apartment with two rooms, one of them set up as a kitchen-living room for the experiments and another area, isolated, for the researchers. In the experiment area, there are three RGBD cameras combined with the NVIDIA Jetson Nano development kit that facilitate the detection and tracking of the pose of people in the environment. The CORTEX architecture for these tests includes all the agents presented in the article and are distributed in different computers with Linux Ubuntu 20.04 distribution and with the RoboComp framework installed (51).

In the experiment area, we use a TV and a table, which are the objects for which time-dependent social interaction spaces are defined. This information is configured offline by the center's staff, who also participate in the event. We recruited 10 participants, of which 8 are non-professional roboticists, and asked them to assume two different situations: i) the therapist interacts with the TV; and ii) the therapy is performed sitting on the table, always chosen by the person in a random position. Each experiment involves one subject and one therapist. We make no assumptions about the accuracy with which the subjects performed the therapy. The robot moved around the kitchen-living room, always taking into account the agenda. Next, participants completed a Likert scale-based questionnaire whose main outcomes are shown in Table 3. In summary, participants reported that they felt comfortable with the robot in the scene. Therapists also responded that the robot was not disturbing during therapy sessions. A sample of these experiments are described in Fig. 22 and Fig. 23. Fig. 22a shows the social mapping in this first test. First, we can observe the Gaussian curves associated with the personal spaces of the participants, labeled as 1 and 2 in Figs. 22a and 23a. In this first test, the therapy takes place at the table; that is why the time-dependent social interaction spaces of the TV are at a light yellow color (lower cost). The planned path is shown in Fig. 22b, where the elastic band is also included. The robot correctly follows the path without disturbing the people participating in the therapy. Fig. 23a illustrates the second test. In this case, the therapist interacts with the TV, while the other participant does not perform the activity yet. For this reason, the robot plans an alternative route and does not cross in front of the therapist (Fig. 23b)[4].

The test described in Fig. 23b was reproduce 10 times to evaluate the performance of the algorithm according to the same variables described in the simulated experiments. The robot always starts and ends its motion in the same places. Results have been compared with an algorithm without social navigation, which obviously does not take into account time dependency when planning the route. The results are presented in Table 4. In this example, the robot performs a more prolonged displacement using our social navigation framework. However, it avoids interrupting the therapist's interaction with the TV. Although most of the time d_t is low, the robot navigates behind the

[4] Third experiment video: `https://youtu.be/RE_g3sHPeGg`

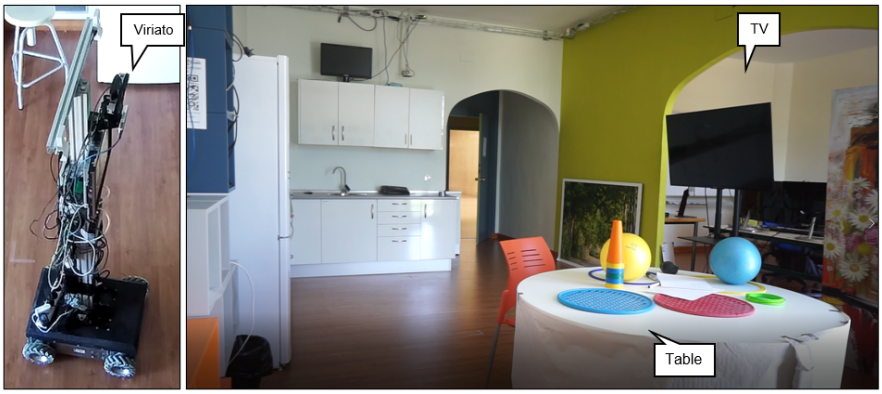
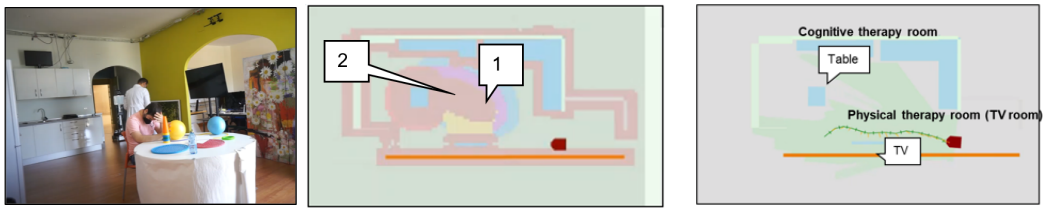
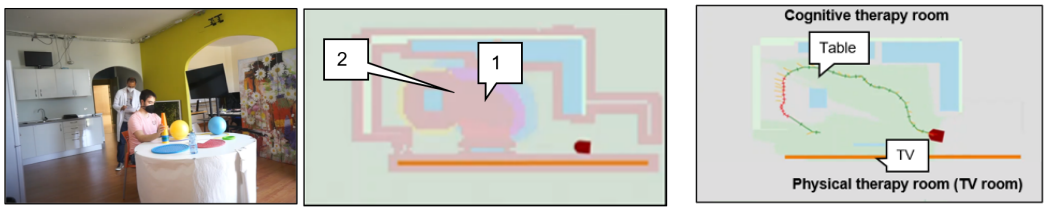


Fig. 21: Left, the robot used for the experiments in real scenarios: Viriato. Right, the laboratory environment.



(a) (b)

Fig. 22: First real test: A person participates in cognitive therapy at the table. a) set-up and social mapping of the real test; and b) path planned and elastic band.



(a) (b)

Fig. 23: Second real test: the therapist is setting up a physical therapy on the TV. a) set-up and social mapping of the real test; and b) path planned and elastic band.

therapist, always keeping away from his personal space. Although this is an initial experiment, in a straightforward environment, the results show that the robot can reach its goal with an acceptable time overhead while ensuring that the social interaction spaces are respected.

7 Conclusions and future works

Human-aware robot navigation is a very complex skill that has to take into account situations involving, among others, human-human, human-robot, or human-object interactions. Getting

Table 4: Navigation results for the real test shown in Fig. 23

Parameter	**Path** Value	**Non social behavior** Value
d_t *(m)*	7.2	4.1
τ *(s)*	58.5	22.1
CHC	8.1	7.2
$d_{min}^{therapist}$*(m)*	0.9 (back)	0.8
$d_{min}^{participant_1}$ *(m)*	1.1	1.5
Ψ *(Intimate) (%)*	0.0	0.0
Ψ *(Personal)(%)*	24.2	8.1
Ψ *(Social)(%)*	10.2	11.3
Ψ *(Public)(%)*	55.6	81.6
Ψ *TV (%)*	0.0	35.2

all this information is not trivial, and it requires algorithms that can continuously detect the people's intentions in the scenario: does this person want to interact with an object? Are these people interacting with each other? Extracting this information from the scenario is complex and challenging to reproduce. Perhaps applying fuzzy logic to infer high-level human intention or activities could be an exciting approach to solving this problem (3). In addition, fuzzy rules could provide a way to incorporate non-trivial social intentions in the design of a SAR.

In the case, for instance, of care facilities, some of these interactions are associated with activities scheduled by the center's professionals. Something similar occurs in other scenarios, such as museums, educational institutions or shopping centers. This simplifies obtaining these intentions by allowing the robots to assume that these spaces are being used at a particular time. Our approach can be understood as a first step towards introducing in the field of robotics navigation, the broad spectrum of social conventions suggested by chronemics. However, an exciting challenge about detecting the time-dependent interactions between persons and objects in real-time arises, furthermore, of how to act in consequence. Obviously, it is much more complicated than acting according to programmed activities, but the proposed underlying architecture has the building blocks to support it.

Consequently, taking into account this time-dependence in the social path planning algorithm is the main novelty of the approach described in this paper. This framework could be applied to many semi-organized social environments, such as caregiving centers. The proposal is based on the well-known Dijkstra's algorithm, where the original free space graph is modified according to time-dependent social interaction spaces. The navigation framework described in this article uses a shared representation of the environment and some CORTEX cognitive architecture agents. The use of CORTEX architecture facilitates the insertion or modification of the software agents in charge of the different functionalities. A social map is initially built from the people and objects in the environment and their possible interactions. As not all these interaction spaces have the same occupation in time, this framework adds temporal information in the planning of the socially accepted path. The validation of the navigation framework has been done in simulated environments, where the tests are easily reproducible. Accordingly, the tests conducted in real environments obtained satisfactory results.

Future work will involve the use of other questionnaires to gather information about the acceptability of the paths performed. Another line of future work consists of including information about the role of people in the center. In this way, the morphology of the personal spaces will vary depending on the *type* of the person, with the aim, for example, that the robot can approach more to the center staff or maintain a more significant social distance with new residents.

Acknowledgment

This work has been partially supported by the Spanish government project RTI2018-099522-B-C42 and by Feder funds and the Extremaduran Government project GR15120 and IB18056, and by the INNOINVEST-4E project (INTERREG V España-Portugal POCTEP).

Declarations

Funding

This work has been partially supported by the Spanish government project RTI2018-099522-B-C42 and by Feder funds and the Extremaduran Government project GR15120 and IB18056, and by the INNOINVEST-4E project (INTERREG V España-Portugal POCTEP)

Competing Interests

All the authors declare no conflict of interest.

Availability of data and material Not applicable.

Code availability Not applicable.

Authors Contributions

Conceptualization, P. Nuñez. and L.V. Calderita; methodology, P. Nuñez and L. V. Calderita; software, A. Vega and P. Bustos; data curation, A. Vega; formal analysis, P. Nuñez; investigation, P. Nuñez and L. V. Calderita; resources, L. V. Calderita and A. Vega; validation, A. Vega; writing–original draft preparation, P. Nuñez and L. V. Calderita; writing–review and editing, P. Nuñez, L. V. Calderita and P. Bustos; supervision, P. Nuñez and L. V. Calderita; project administration, P. Nuñez; funding acquisition, P. Nuñez and P. Bustos.

Ethics Approval

Not applicable.

Consent to Participate

Not applicable.

Consent to Publish

All the authors consent to the publication of the manuscript.

References

1. P. Flandorfer. Population Ageing and Socially Assistive Robots for Elderly Persons: The Importance of Sociodemographic Factors for User Acceptance. International Journal of Population Research, 2012.
2. Calderita, L. V., Vega, A., Barroso-Ramírez, S., Bustos, P., and Núñez, P. Designing a Cyber-Physical System for Ambient Assisted Living: A Use-Case Analysis for Social Robot Navigation in Caregiving Centers. Sensors, 20(14), 4005. (2020)
3. Kuo, I. H., and Jayawardena, C. Application of fuzzy techniques in human-robot interaction-a review. In International Conference on Social Robotics. Springer, Cham, 249-255 (2014).
4. Palm, R., Chadalavada, R. and Lilienthal, A.Fuzzy Modeling and Control for Intention Recognition in Human-robot Systems. Proc. of the 8th Intern. Joint Conf. on Comp. Intell. (IJCCI 2016) - Volume 2: FCTA, 67-74 (2016)
5. M. Beetz and B. Johnston B. International Conference on Social Robotics. Lecture Notes in Computer Science, Williams MA, vol 8755, Springer, 2014.
6. A. Vega, L.J. Manso, P. Bustos and P. Núñez. Planning Human-Robot Interaction for Social Navigation in Crowded Environments. Proceedings of Workshop of Agentes Físicos. Madrid, 2018.
7. Hall, E. Proxemics. Current Anthropology, vol. 9, no. 2-3, pp. 83108, 1968.
8. K. Charalampous, I. Kostavelis and A. Gasteratos. Robot navigation in large-scale social maps: An action recognition approach. Expert Systems with Applications. Vol 66. pp 261 –273. 2016. Elsevier.
9. J. Rios-Martinez. Socially-Aware Robot Navigation: combining Risk Assessment and Social Conventions. Hal.Inria Frace, 2013.
10. Bruneau, T. J. Chronemics and the verbalnonverbal interface.), The relationship of verbal and nonverbal communication, pp. 101-–117, 1980.
11. Calderita, L.V. Deep State Representation: an unified internal representation for the robotics cognitive architecture CORTEX. PhD thesis, Universidad de Extremadura, 2016.
12. A. Vega, R. Gondkar, L. Manso and P. Núñez. Towards efficient human-robot cooperation for socially-aware robot navigation in human-populated environments: the SNAPE framework. International Conference on Robotics and Automation, 2021.
13. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Numerische mathematik, 1, 269–271 (1959).
14. Hart, P. E., Nilsson, N. J., and Raphael, B. A formal basis for the heuristic determination of minimum cost paths. IEEE transactions on Systems Science and Cybernetics, 4(2), 100-107 (1968).
15. Stentz, A. The D* Algorithm for Real-Time Planning of Optimal Traverses. CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, (1994).
16. Althaus, P., Ishiguro, H., Kanda, T., Miyashita, T., Christensen, H.I.: Navigation for Human-Robot Interaction Tasks. IEEE International Conference on Robotics and Automation. Volume 1, pp. 1894, 1989, (2004).
17. P. Scales, O. Aycard and V. Aubergé. Studying Navigation as a Form of Interaction: a Design Approach for Social Robot Navigation Methods. In IEEE International Conference on Robotics and Automation (ICRA), (2020).
18. T. Kruse, A. Pandey, R. Alami and A. Kirsch. Human-aware robot navigation: A survey. Robotics and Autonomous Systems. Vol 61. n 12. pp 1726–1743. 2013. Elsevier.
19. J. Rios-Martinez, A. Spalanzani, and C. Laugier. From proxemics theory to socially-aware navigation: A survey. International Journal of Social Robotics. Vol 7. n2. pp 137–153. 2015. Springer.
20. K. Charalampous, I. Kostavelis and A. Gasteratos. Recent trends in social aware robot navigation: A survey. Robotics and Autonomous Systems, 92, 2017.

21. T. Thoa, C. Copot and R. De Keyse. Heuristic approaches in robot path planning: A survey. Vol. 86, pp 13–28, 2016.
22. M. Costa and M. Silva. A Survey on Path Planning Algorithms for Mobile Robots. 19th IEEE International Conference on Autonomous Robot Systems and Competitions, Portugal, 2019.
23. D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. IEEE Robotics and Automation, 4(1), 1997.
24. O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. Int. J. Rob. Res., 5(1):90–98, April 1986. ISSN 0278-3649
25. Quinlan, S. and Khatib, O. Elastic bands: connecting path planning and control. Proceedings IEEE International Conference on Robotics and Automation, Atlanta, GA, USA, pp. 802-807, vol.2, (1993) doi: 10.1109/ROBOT.1993.291936.
26. J. den Berg, M. Lin and D. Manocha. Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2008.
27. D. Helbing and P. Moln´ar. Social force model for pedestrian dynamics. Phys. Rev. E, 51:4282–4286, May 1995.
28. K. Cai, C. Wang, J. Cheng, S. Song, C. de Silva and M. Meng. Mobile Robot Path Planning in Dynamic Environments: A Survey. In arXiv:2006.14195.
29. Kirby, R., Simmons, R., Forlizzi, J.: COMPANION: A Constraint-Optimizing Method for Person-Acceptable Navigation. IEEE International Symposium on Robot and Human Interactive Communication, (2009).
30. Tranberg Hansen, S., Svenstrup, M., Andersen, H.J., Bak, T.: Adaptive Human Aware Navigation Based on Motion Pattern Analysis. IEEE International Symposium on Robot and Human Interactive Communication, (2009).
31. E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Simeon. A human aware mobile robot motion planner. IEEE Transactions on Robotics, 23(5), 2007.
32. H. Kivrak, P. Uluer; H. Kose, E. Gumuslu, D. Erol-Barkana, F. Cakmak and S. Yavuz. Physiological Data-Based Evaluation of a Social Robot Navigation System, in 29th IEEE International Conference on Robot and Human Interactive Communication, (2020).
33. C. Tsai and J. Oh. A Generative Approach for Socially Compliant Navigation, in IEEE International Conference on Robotics and Automation, (2020).
34. Y. Che, A. M. Okamura and D. Sadigh. Efficient and Trustworthy Social Navigation via Explicit and Implicit Robot–Human Communication IEEE Transactions on Robotics, Volume: 36, Issue: 3, pp 692-707, (2020).
35. C. Yang, T. Zhang. L. Chen; L. Fu. Socially-Aware Navigation of Omnidirectional Mobile Robot with Extended Social Force Model in Multi-Human Environment. In IEEE International Conference on Systems, Man and Cybernetics, 2019
36. F. Pimentel and P. Aquino. Proposal of a New Model for Social Navigation Based on Extraction of Social Contexts from Ontology in Service Robots, in IEEE Latin American Robotics Symposium, 2019
37. A. Vega, L.J. Manso, P. Bustos, P. Núñez D.G. Macharet. Socially Aware Robot Navigation System in Human-populated and Interactive Environments based on an Adaptive Spatial Density Function and Space Affordances. Pattern Recognition Letters. Vol. 1, Pages 72-84, 2019
38. Weihua C., Tie Z., and Yanbiao, Z.: Mobile robot path planning based on social interaction space in social environment. International Journal of Advanced Robotic Systems. Volume 1, pp 1–10, 2018.
39. M. Kollmitz, K. Hsiao, J. Gaa, and W. Burgard. Time dependent planning on a layered social cost map for human-aware robot navigation. In European Conference on Mobile Robotics. 1-6. 2015.
40. A. Bera, T. Randhavane, R. Prinja and D. Manocha. SocioSense: Robot Navigation Amongst Pedestrians with Social and Psychological Constraints. IEEE/RSJ International Conference on

Intelligent Robots and Systems (IROS) September 24–28, 2017, Vancouver, BC, Canada
41. A. Bera, T. Randhavane, R. Prinja, K. Kapsaskis, A. Wang, K. Gray and D. Manocha The Emotionally Intelligent Robot: Improving Social Navigation in Crowded Environments. IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019.
42. Kostavelis, I. Robot Behavioral Mapping: A Representation that Consolidates the Human-robot Coexistence. Robotics and Automation Engineering Journal. Vol. 1, 2017.
43. Haut, M., Manso, L., Gallego, D., Paoletti, M., Bustos, P., Bandera, A., and Romero-Garcés, A., 2016. A navigation agent for mobile manipulators, in: Robot2015: Second Iberian Robotics Conference, Springer. pp. 745–756.
44. L. Manso, L. V. Calderita, P. Bustos, and A. Bandera. Use and advances in the active grammar-based modeling architecture. Journal of Physical Agents2016, pp. 33–38.
45. S. Satako, T. Kanda, D. Glas, M. Imai, H Ishiguro and N. Hagita. How to approach humans?-strategies for social robots to initiate interaction, 4th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 109-116, 2009.
46. L. Manso, P. Bustos, R. Alami, G. Milliez, and P. Núnez. Planning human-robot interaction tasks using graph models. In Proceedings of International Workshop on Recognition and Action for Scene Understanding (REACTS 2015), pp. 15-27.
47. Wanko Keutchafo, E.L., Kerr, J. and Jarvis, M.A. Evidence of nonverbal communication between nurses and older adults: a scoping review. BMC Nurs 19, 53 (2020). https://doi.org/10.1186/s12912-020-00443-9
48. Laamarti, F., Eid, M., and El Saddik, A: An overview of serious games. International Journal of Computer Games Technology, (2014).
49. Okal, B., Arras, K. Learning socially normative robot navigation behaviors with bayesian inverse reinforcement learning. IEEE International Conference on Robotics and Automation, pp 2889 – 2895, (2016).
50. Kostavelis, I. Robot Behavioral Mapping: A Representation that Consolidates the Human-robot Coexistence. Robotics and Automation Engineering. Volume 1, pp 1–3, (2017).
51. Manso, L. J., Bachiller P., Bustos, P. Núñez, P., Cintas, R., and Calderita, L. V. Robocomp: a tool-based robotics framework. Lecture notes in computer science. simulation, modeling and programming in autonomous robots. Springer, 2010, vol. 6472, p. 251–262.