Abstract—This paper presents an approach to task planning under uncertainty. The main goal is to attain optimal human-aware plans, by optimizing an accumulated discounted reward over a finite time horizon, computed using decision-theoretic methods, and executed by a Networked Robot System (NRS) in an indoor space. A NRS is a network of autonomous robots and other devices that must be capable of selecting and executing their own actions, in the presence of several events occurring in their surrounding environment. Events are detected with different levels of confidence, due to uncertainty in sensing. Robot’s actions have different levels of uncertainty concerning their impact in the world. ROS packages are used to implement actions and perceptions of the NRS. To achieve planning under uncertainty, a Partially Observable Markov Decision Process (POMDP) is applied to a task. POMDPs are a mathematical framework for sequential decision-making in partially observable environments with Markovian transition models and additive rewards. The task developed corresponds to a prevention and safety action where the NRS should be capable of warning humans that they may be moving or staying at a restricted or dangerous location and should not be at that place. Experiments evidence that the model can be used to accomplish the objectives and that it is possible to perform human robot interaction in real world environments.

I. INTRODUCTION

Technological progress in the last few decades allowed the evolution of scientific fields such as computation, networked vehicles, communications and artificial intelligence. Robots capabilities have been expanding and, while there are still limitations in the usage of robots (e.g., batteries, mobility, equipment), their presence and importance in society is increasing. Robots can now interact with human beings and other robots in cooperative ways. These recent technology systems are denominated Networked Robot Systems (NRSs). A NRS is a network of devices, with sensors and actuators, as well as mobile robots, capable of making autonomous decisions to complete tasks [1]. To make autonomous decisions, a NRS acquires information about the environment, using sensors, and executes actions according to the information received and its goals, using actuators. These systems are connected to a communications network that are wired or wireless.

One of the main objectives and purposes of creating NRSs is enabling Human-Robot Interaction (HRI). The number of applications for robots in the real world has increased in the past and continues to expand in the present. Applications include manufacturing, interaction with human beings in tasks of guidance, personal assistance, transportation of goods, elderly care, providing information, rescue operations, bomb diffusion, surveillance, monitoring and entertainment [2].

The successful completion of a task depends on the robots decisions and actions. Actions are chosen according to the robot’s environment, which may be stochastic. Robot’s sensors are often limited to observing its direct surroundings. Furthermore, robot’s sensors may contain technical limitations or imperfections (e.g. range, resolution, noise) which are another cause of uncertainty. Robot’s actuators involve motors that are affected by noise, legs or wheels that may not be able to roam in certain locations and arms which are limited to certain movements because of its joints. Since NRSs are real-time systems, computation carried out is limited which may decrease the accuracy of algorithms used.

Planning is made in a model of the world. Since models are abstractions of the real world, not all effects of the real world are modelled. Therefore, state, observation and action models are not completely accurate. Human presence in environments increases the chance of inaccuracy in modelling, since their intentions are difficult to predict [2]. Human-aware task planning under uncertainty is an important research topic in robotics. It consists of designing and implementing optimal plans, subject to uncertainties regarding action effects and sensor observations, to perform HRI. Optimal planning is what makes robots capable of selecting and executing actions on their own. Every time the agent acts, it receives a reward associated with the action taken and the state of the environment. The goal is to achieve an optimization of the accumulated discounted reward over a finite time horizon. To overcome the problem of uncertainty, the Partially Observable Markov Decision Process (POMDP) is used.

To validate human-aware planning, a prevention and safety task capable of warning people to avoid moving or staying in a restricted/dangerous place was created. A POMDP model was developed for this purpose. The model assumes navigation actions that were implemented in a robot. An observation system, composed by a robot and a surveillance camera, enabling to observe the position and movement of a person was developed. The POMDP observations models were conceived based on the accuracy of this observation system. Experiments were conducted to test the implemented model at the Laboratório de Robótica Móvel (LRM) at Institute for Systems and Robotics (ISR). Results showed that the robot is capable of moving to restricted/dangerous places when a person is observed going (or staying) to the exact same place, in order to avoid that person from going (staying) there. These results validated the usage of a POMDP in uncertainty conditions.
The paper is structured as follows: section II presents related work in planning under uncertainty; section III introduces a background with concepts of the POMDP mathematical framework; section IV refers to an HRI task, modelled into a POMDP, where uncertainty is involved and decision-making is required; section V covers the implementation and resolution of the designed model and how actions are executed and observations perceived by the NRS; section VI exhibits the experiments conducted and their results; section VII concludes the paper with remarks on the work developed and possibilities/suggestions for future work.

II. RELATED WORK

Interactions between robot systems and people have been increasing over time. Survey [3] shows that robots can be used in health care and therapy as participants showed reduced levels of stress and better social interaction. Survey [4] demonstrates that robots can be used in areas such as autism; elderly care; intelligent wheelchairs; assistive robotic arms; prosthetic limbs; and post-stroke rehabilitation. In [5], therapeutic seal robots were introduced in an elderly care house to evaluate the effects of interactions between robots and residents. Residents displayed a better mood, lower stress levels and became more active and communicative with each other and with caregivers. Patients with dementia showed improvements in their cortical neurons activity.

Two major projects with NRSs are: the URUS project which focused on developing a network of robots that would be able to interact with human beings for tasks of assistance, transportation of goods and surveillance in urban areas; and the Japan NRS project, which has the purpose of providing information and support to people [1].

Applications of POMDPs in real world situations have been studied over time and have been of great importance in different areas. In [6], a POMDP model is used on tasks relevant to support elderly people with cognitive or physical disabilities. The tasks are: assisted hand washing, health and safety monitoring, and wheelchair mobility. In [7], an approach is made to find the best course of action to assist people with dementia in a task of handwashing. A POMDP model is used to deal with partial observability and plan when and how assistance is needed in each situation. The model was solved with Perseus-ADD. In [8], it is shown how to model a cooperative perception task, of tracking and classifying people, as a POMDP in a NRS. Results illustrated that it is possible to trade the cost of moving a robot with desired level of confidence regarding an event. The Nursebot project [9] was created to help nurses and improve the daily quality of life of elderly people. The project uses autonomous mobile robots equipped with navigation and interaction sensors. In [10], robot navigation under uncertainty is performed. It is studied how a point-based algorithm can help computing POMDPs to successfully perform navigation in 2D and 3D environments.

III. BACKGROUND

A. Partially Observable Markov Decision Process

A POMDP is a mathematical model for sequential decision-making in partially observable environments. The process assumes the existence of an agent that interacts with the environment. In a POMDP the agent is unable to directly observe states, observing instead possible sensor readings. A POMDP model can be described as:

- a finite set of states \( S \);
- a set of actions \( A(s) \) for each state \( s \);
- a transition model \( T(s,a,s') = P(s'|s,a) \) that corresponds to the probability that action \( a \) in state \( s \) will lead to state \( s' \);
- a reward function \( R(s,a) \) which is the immediate reward received for taking action \( a \) when in the current state \( s \);
- a discrete set of observation \( \Omega \);
- an observation model \( O(s',a,o) = P(o|s',a) \) representing the probability of observing \( o \) in state \( s' \) after executing action \( a \).

The next state \( s' \) depends on the current state \( s \) and the chosen action \( a \). After the transition between states, the agent receives a reward \( r \) and perceives an observation \( o \) providing information from the state \( s' \).

![Figure 1. Decision network representing a POMDP model.](image)

The goal in optimal planning is to choose a policy (a specification of the agent’s behaviour for any state it might reach) that allows the agent to act in a way as to maximize a cumulative function of expected long-term reward,

\[
E \left[ \sum_{t=0}^{h} \gamma^t R(s_t, a_t) \right],
\]

where \( E[\cdot] \) denotes the expectation operator, \( h \) is the planning horizon and \( \gamma \) is a discount factor that satisfies \( 0 \leq \gamma < 1 \).

In a POMDP, the agent’s past behaviour is not stored in memory. Instead, the agent is in what is called a belief state \( (b) \) which is the set of possible states. All possible states in the environment are considered, which means that the belief state is a probability distribution over all states. This probability distribution allows to keep the same amount of information as saving the agent’s behaviour over time would [11].
The belief state is updated by the Bayes rule [12],
\[
    b'(s') = \frac{O(s', a, o)}{P(o|a, b)} \sum_{s \in S} T(s, a, s')b(s),
\]
(2)
where \(P(o|a, b)\) is a normalizing constant with \(P(o|a, b) = \sum_{s' \in S} O(s', a, o) \sum_{s \in S} T(s, a, s')b(s)\).

To choose actions, that enable to complete a task in the best possible way, it is necessary to select them according to the long-term effects on the agent’s total reward [13]. Accomplishing this means finding the optimal policy \(\pi^*(b)\) for the agent. The quality of a policy is measured by an expected utility value function \(V^*(b)\). The value function is defined as the expected future discounted reward the agent can collect by following a policy \(\pi\) and starting in a belief \(b_0\).

\[
    V^*(b_0) = E \left[ \sum_{t=0}^{h} \gamma^t R(b_t, \pi(b_t)) | b_0, \pi \right]
\]
(3)

The optimal policy is obtained by optimizing the long-term reward.

\[
    \pi^*(b) = \arg\max_{\pi} V^*(b_0)
\]
(4)

Knowing the optimal policy, the highest expected reward value for each belief state, that is, the Bellman optimality equation is given by,

\[
    V^*_h(b) = \max_{a \in A} \left[ r(b, a) + \gamma \sum_{o \in O} O(b, a, o)V^*_{h-1}(b') \right],
\]
(5)
where \(r(b, a) = \sum_{s \in S} b(s)R(s, a)\).

The optimal value function is continuous, piecewise linear and convex. The value iteration method builds a sequence of value function estimates which tends to the optimal value function for the current task [14]. A value function in a finite-horizon can be parameterized by a finite number of vectors \(\alpha\) over the belief space. Each vector increases the value function in a certain region. Each vector has an associated action, which is the optimal one for beliefs in that region [15]. Figure 2, adapted from [15], shows an example of how a belief space may be divided into regions, each with an associated vector.

\[
    V_{n+1} = \tilde{H}_{\text{PERSEUS}} V_n,
\]
(6)

making certain that \(V_{n+1}(b) \geq V_n(b), \forall b \in B\).

To accomplish the goal of human-aware planning under uncertainty, a model of a POMDP task was developed.

**A. Task**

The objective of the developed task is for an agent, in this case an NRS constituted by a robot and a surveillance camera, to predict the behaviour of people and act accordingly to keep them from moving or staying in a specified location. A person moves freely in an environment which contains restricted/dangerous areas where the person should not move into. When a person moves to one of those areas or evidences the intention to, the robot should move to the corresponding place and verbally advise the person that the place is restricted/dangerous and should not be visited. A person shows intention of moving to a place, by continuously moving towards it. If a person is moving at a higher speed,
it is also more likely to be moving somewhere. This means that the robot needs to predict the behaviour of each person to a degree of certainty that allows the robot to know how to act. It is highly relevant to note that when no person is near a restricted/dangerous area but the robot is far from every restricted/dangerous spot, the robot should navigate to an equidistant position from all restricted/dangerous areas. This would enable the robot to reach a restricted/dangerous place faster and inform people that they should not be or go there.

This task has multiple real world applications in indoor environments such as: preventing children from moving or staying in dangerous places thus providing assistance to adults in charge of supervising children; or preventing adults, in particular elderly people who may not be in possession of all faculties, from entering restricted or hazardous places in hospitals such as radiation or operating rooms.

B. General model

The POMDP model, Figure 3, of the task considers \( n \) people and \( r \) restricted/dangerous areas. The model contemplates two state variables for every person, these two variables being the position and movement direction of each person. The model also takes into account the position of the robot and its navigation actions. Finally, both the robot’s and the people’s position as well as the people’s velocity are observed and interpreted by the model. The general framework has the following terms:

- a state variable for each person’s movement direction, \( D_1, ..., D_n \);
- a state variable for each person’s position, \( P_1, ..., P_n \);
- a state variable for the robot’s position, \( R \);
- an action variable containing the possible navigation actions for the robot to perform, \( A_{\text{Move}} \);
- a reward function, where rewards are assigned according to the robot’s actions, position and people’s behaviour;
- an observation variable for each person’s velocity, \( O_{V1}, ..., O_{Vn} \);
- an observation variable for each person’s position, \( O_{P1}, ..., O_{Pn} \);
- an observation variable for the robot’s position, \( O_R \).

Velocity is a physical vector quantity that is defined by two factors: magnitude and direction. Velocity’s magnitude corresponds to speed, while direction corresponds to the direction of motion.

1) States and transitions: The environment’s map contains \( p \) positions where a person and the robot may be. The possibility of a person or the robot not being in the map is also considered. Positions are seen as cells of the map. Cell boundaries are defined using two dimensional Cartesian coordinates \((x, y)\). A person may be moving in one of the four cardinal directions or not moving. Table I presents the model’s state variables and corresponding possible states.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Robot ((R))</th>
<th>Person Position ((P))</th>
<th>Person Direction ((D))</th>
</tr>
</thead>
<tbody>
<tr>
<td>States</td>
<td>1, ..., (p), Out of the map</td>
<td>1, ..., (p), No person</td>
<td>North, West, East, South, Not moving</td>
</tr>
</tbody>
</table>

The transition model for the robot’s position depends on two factors, the robot’s previous belief state and the action taken. Since the actions are navigation actions, each time an action is carried out, the robot may change its position in the environment. Assuming the policy is to move, the next state will likely (higher probability) be the one to where the robot should navigate. If the robot moves to east, for example, its position will likely change to the adjacent east position. There is also the possibility that the robot is unable to move due to an obstruction in the environment or due to a problem with its actuators. In that case, the robot will stay in the same position. The probability of the robot staying in the same position when the policy is to move is low. If the action required is not to move, the robot stays in the previous state.

The transition model of the person’s moving direction only takes into account the previous belief state. A person maintaining the same movement direction between steps is more likely. There is still a relatively high probability of changing direction. The lowest transition probability consists of the person moving in the opposite direction in the next step.

The transition model of the person’s position considers the previous belief state of that variable and the moving direction. If a person is not moving then it is likely that they will stay on the same position. If a person is moving towards a certain direction, there is a higher probability that they will either stay on the same position or on a position located in that direction with adjacent positions being of higher probability. If a person moves to a direction where there is a wall, it is more likely that they will either stay on the same position or on an adjacent position.

State variables of different people do not influence each other, meaning that the behaviour of a person has no impact on another one.

2) Actions: Actions are important since they enable an agent to reach its goal and complete tasks. The possible actions of the model are confined in one variable called \( A_{\text{Move}} \) which is presented in Table II.
Only navigation actions are considered since the task is a navigation task where the robot needs to prevent a person from entering an area. Four actions help the robot to move to a desired place while one action allows the robot to avoid moving. The navigation actions are named as the four cardinal directions representing the direction to where the robot should move. So, if the generated policy is "West", the robot should move to the west adjacent position. The navigation actions are designed to make the robot move continually through out the environment instead of moving directly to the restricted/dangerous area. This happens because the person may decide not going into that area and in this case the robot can stop moving towards it.

In case multiple people are considered in the model and multiple people move to different restricted/dangerous places, the robot will proceed to the area that is either closest or more likely to be visited by someone. Finally, it is important to avoid the robot leaving the environment. This is done by giving a penalty with a high absolute value, if the robot is ever outside of the map (by being on an edge of the map and exceeding that edge in a navigation action). The negative reward of leaving the map avoids this situation. The overall behaviour ensures a higher reward in the long term.

4) Observation models: If a variable is in a specific state, it is more probable to receive an observation that corresponds to that state. At each time step, the belief state for each variable changes according to the transition model and the observation received. There are two observation models for each person, one for position and one for velocity, and one observation variable for the robot’s position. Table III displays the model’s observation variables and corresponding possible observations.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Obs Robot (O_R)</th>
<th>Obs Person Position (O_P)</th>
<th>Obs Person Velocity (O_V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observations</td>
<td>1, ..., p. Out of the map</td>
<td>1, ..., p. No person</td>
<td>v_1, ..., v_p. Not moving</td>
</tr>
</tbody>
</table>

In the case of the robot’s position, the observation is deterministic. That is because the robot always moves in the direction where it should move, or stays in the same position. This behaviour is in fact inserted in the robot’s transition model. It is impossible for the robot to be observed in a place which was not its target when it started navigating. The navigation system implemented in the robot ensures that the robot moves to the supposed goal unless the robot cannot do it due to a possible obstruction or a failure in the actuators.

The observation model for the position of a person considers that if a person is in a given state (position), there is a high probability that the observation received is being in that state. However, it exists a low probability that the person is seen in a different position, resulting from bad readings in sensing or false positives in people detection. If they are seen in a different position, it is more likely to be near the correct one than far away. If there is no one in the environment, there is a high probability that no one is detected. There is still the possibility of a person being detected due to the existence of false positives.

Observations of a person’s velocity contain one of the four cardinal directions and a speed. Speed is discretized in classes. The observation model for a person’s velocity considers that, if a person is moving towards a direction, it is more likely that they are moving at a higher speed. When someone is moving to a place, they usually walk at a high and constant speed. An observation of a person moving towards a direction, decreases the belief that they are moving to a different one, particularly the opposite direction. So, if a person’s movement direction state displays a certain direction, the probability of moving in that direction is higher than others, and the probability of
moving at a higher speed is also higher. If no person is detected or if a person is not moving, it is more likely observing no movement. When a person is not moving, the probability of moving at a certain speed is low. As speed increases, the lowest is the probability.

V. IMPLEMENTATION

A. NRS and indoor environment

The NRS used in experiments consists of a robot called MBot and a surveillance AXIS camera. The indoor environment where HRI experiments were conducted is the LRM, at ISR. The robot moves through an open space of the lab and the surveillance camera is positioned on the lab’s ceiling and is used to observe that same space. The robot needs to have previous knowledge of the environment in order to navigate. This can be achieved using a 2-D occupancy grid map of the environment. The map was created from laser and pose data collected by the mobile robot. Only a part of the lab was used in the experimental work.

![Figure 4. 2-D occupancy grid map of LRM. The box represents the section of the lab (environment) where experiments occurred.](image)

B. Case study model

The case study model considers one person in the environment, \( n = 1 \), and twelve positions, \( p = 12 \). The twelve positions consist of twelve cells that divide the environment as in Figure 5. When a person is in a position that belongs to a cell, that person is seen as being in that cell. The cells are numbered from 1 to 12 and therefore the person’s and robot’s position, if in the map, is also indicated as being between 1 and 12. The robot navigates between waypoints, positioned in the map, each inside a cell.

![Figure 5. Cell division of the environment. Each cell represents a region of the map where a person may be present.](image)

Two versions of the same POMDP model were developed. One with one restricted/dangerous area in the environment, \( r = 1 \), and the other with two areas, \( r = 2 \). The difference between the two versions stands only on the reward functions since it is the reward function the one responsible in assigning a purpose to certain areas. When there is only one restricted/dangerous area, that area corresponds to the region of cell number 1. In case of two restricted/dangerous areas, these areas correspond to cells number 1 and 12. In both cases, the region of cell 7 is the equidistant/resting spot where the robot should be when no person displays hazardous behaviour. The directions in the environment correspond to the orientation in the map in Figure 4 and therefore also to the orientation in Figure 5.

The velocity observation variable has both speeds and directions. Direction is one of the four cardinal directions. Speed is one of three possible speed classes: between 0 and 0.2 meters per second; between 0.2 and 0.7 meters per second; and over 0.7 meters per second. There is also the possibility of the person not moving or not being in the environment, in both cases an observation of “Not moving” is made.

For every navigation movement, the agent receives a reward of \(-0.2\), in order to avoid unnecessary navigation. Other values of rewards for specific situations are presented in Table IV.

<table>
<thead>
<tr>
<th>Robot</th>
<th>Person’s Behaviour</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>At the restricted cell</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>At an adjacent cell from the restricted one and not moving away from it</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>At a distance of two cells from the restricted one and moving towards it</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Different from the previous three</td>
<td>(-10)</td>
<td></td>
</tr>
<tr>
<td>Out of the map</td>
<td>(-30)</td>
<td></td>
</tr>
<tr>
<td>Other cells</td>
<td>(-3)</td>
<td></td>
</tr>
</tbody>
</table>

Table IV: REWARD ASSIGNMENT ACCORDING TO THE ROBOT’S POSITION AND THE PERSON’S BEHAVIOUR. LINES SYMBOLIZE CASES WHERE THE PERSON’S BEHAVIOUR HAS NO INFLUENCE ON REWARDS.

The agent receives negative rewards at most positions. This forces the robot to move to the equidistant place. However, when a person is seen as moving to a restricted/dangerous region or is already there, the robot will also move to that region. While the agent will receive penalties for not being at the equidistant spot nor a restricted/dangerous place and for moving, once it reaches the desired place and alerts the person, the agent will receive a high positive reward.

In the POMDP model, the chosen value for the discount factor \( \gamma \) is 0.99. Therefore, the cumulative function of rewards becomes,

\[
E \left[ \sum_{t=0}^{h} 0.99^t R(s_t, a_t) \right],
\]

where \( E[\cdot] \) denotes the expectation operator and \( h \) is the planning horizon.
C. Integration

Symbolic Perseus is the solver used to perform offline planning on the developed POMDP. Once the model has been solved, it is possible to evaluate the policy graph generated. This can be achieved with the evalPOMDPpolicy function granted by the MATLAB code of Symbolic Perseus. The function simulates belief states for all variables and executes the optimal action in each scenario. Finally, actions are executed with the tracePOMDPpolicy function. The function receives observations inserted manually and outputs policies.

The overall system used in experiments with the NRS can be divided into three subsystems: information perceived as observations; the solver Symbolic Perseus which provides decisions on how to act at each time step; and the execution of actions by the NRS. A description of the integrated system is displayed in Figure 6.

Communication of informations between the subsystems is provided by ROS topics. Observations are perceived by the the NRS and published in four topics. These topics are subscribed in MATLAB and their content is converted to string format. Strings for movement direction and speed are concatenated thus forming the velocity observation. The tracePOMDPpolicy function receives the observation strings and outputs a policy that is published in a topic. Its content enables robot navigation.

D. Acting

Actions are executed according to a navigation system existing in the robot. The robot moves between cells in the map, each cell containing a waypoint to where the robot should move. The waypoint is defined by the \( x, y \) coordinates in the world frame. The navigation system receives information about map data, the robot’s pose in the map and sensor readings. From that data, it builds a costmap with information about obstacles in the world. A global planner using the \( A^* \) search algorithm finds the optimal path between the robot’s current position and the navigation goal. Knowing the costmap and the global plan, a local planner yields velocity commands that should be executed by the robot. This results in a trajectory for the robot. As the robot moves, the costmap is updated and so is the global navigation plan. When the robot reaches a restricted/dangerous area, it informs the person verbally that they should not be there.

E. Observing

Observations are received from MBot’s onboard laser scanner and from the surveillance camera. The laser is used to detect people’s speed and for the robot’s self localization. In the robot’s self localization, odometry is also used. The camera is used to detect the position of people and to check their moving direction.

When the robot is moving, it uses the AMCL particle filter to check its position in the environment. During navigation, readings given by the laser scanner and the distance covered over time given by odometry, update the particles position estimating the robot’s pose in the known map. While this contains an associated localization error, this error is quite small. Consequently, while the actual position may not be the waypoint’s exact one, the robot will stay inside the correct cell once it finishes its navigation action. This makes the robot’s observation deterministic.

A person’s position can be achieved by perceiving images of the environment and then distinguish people in those images. Images are extracted from the surveillance camera’s video feed. The detection of a person is done using Histogram of Oriented Gradients (HOG) with a trained Support Vector Machine (SVM) [19]. The computation of HOG descriptors occurs according to the upcoming list of steps:

- Evaluate pixels of the image to perform colour normalization and gamma correction;
- Calculate horizontal and vertical gradients of the image;
- Compute histograms of gradients with spatial and orientation binning;
- Select descriptor blocks and perform block normalization;
- Define an over detection window;
- Use a supervised learning classifier.

From the output of the HOG with a trained SVM, the coordinates of the person in the image are known. The image position is converted to the map referential using an homography. The person’s map coordinates belong to one of the twelve regions of the environment and that region corresponds to the person’s cell position.

Movement direction of a person consists of comparing the person’s position (map coordinates, not cell position) at two consecutive moments and finding the correspondent direction vector. By comparing map coordinates in consecutive moments, it is possible to check if the current coordinates are north, west, east or south to the previous ones. Speed is observed using the robot’s laser reading to detect legs. Leg positions are determined over time making it possible to see the distance covered by a person over time. When a person does not move or is not detected in two consecutive video frames, the observation perceived is of no movement.
VI. EXPERIMENTS

In the experiments conducted\textsuperscript{1,2}, the agent was put in multiple situations and its behaviour was analysed as observations were perceived. This makes possible to check if the task is achieved.

A. Simulation with $r = 1$

The first experiment presented considers a simulation in the MATLAB environment. Observations were manually input and the evolution of belief states and the selected actions were studied. A simulation is advantageous as it allows to check if the model behaves as intended in an easily controlled scenario. The experiment considers that there is only one restricted/dangerous place, which is cell 1. Figure 7 and Table V contain the results of the experiment.

![Figure 7. Simulation with one restricted/dangerous area: positions and cumulative reward over time steps.](image)

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Obs Robot</th>
<th>Obs Person</th>
<th>Obs Velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Do nothing</td>
<td>7</td>
<td>9</td>
<td>East higher</td>
</tr>
<tr>
<td>2</td>
<td>Do nothing</td>
<td>7</td>
<td>7</td>
<td>East higher</td>
</tr>
<tr>
<td>3</td>
<td>East</td>
<td>5</td>
<td>5</td>
<td>East higher</td>
</tr>
<tr>
<td>4</td>
<td>East</td>
<td>3</td>
<td>3</td>
<td>East 0-0.2</td>
</tr>
<tr>
<td>5</td>
<td>East</td>
<td>1</td>
<td>1</td>
<td>Not moving</td>
</tr>
<tr>
<td>6</td>
<td>Do nothing</td>
<td>1</td>
<td>4</td>
<td>West 0.2-0.7</td>
</tr>
<tr>
<td>7</td>
<td>Do nothing</td>
<td>1</td>
<td>6</td>
<td>West higher</td>
</tr>
<tr>
<td>8</td>
<td>West</td>
<td>3</td>
<td>8</td>
<td>South 0-0.2</td>
</tr>
<tr>
<td>9</td>
<td>Do nothing</td>
<td>3</td>
<td>7</td>
<td>Not moving</td>
</tr>
<tr>
<td>10</td>
<td>West</td>
<td>5</td>
<td>7</td>
<td>Not moving</td>
</tr>
<tr>
<td>11</td>
<td>West</td>
<td>7</td>
<td>10</td>
<td>North 0-0.2</td>
</tr>
<tr>
<td>12</td>
<td>Do nothing</td>
<td>7</td>
<td>10</td>
<td>Not moving</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table V: Simulation with one restricted/dangerous area: policies and observations over time steps.

The person starts at 11 and moves continuously to east. This increases the belief that the person’s moving direction is east. The belief that the person’s position is the one seen is high since the person moves to east over time steps and is observed in eastern positions over time. Once the person reaches 7, the robot starts to move to the restricted/dangerous area. The person and the robot reach 1 simultaneously. Once the robot informs the person that they should not be there, the person leaves. The robot returns to the resting spot when there is a high belief that the person is not moving to or is near 1.

B. Experiment with $r = 1$

The second experiment with one restricted/dangerous area was performed using the actual robot and the surveillance camera. The person’s behaviour was similar to the one in the simulation. The initial conditions are however different. The robot starts at 8 and the person is not seen in the environment. Figure 8 and Table VI contain the results of this experiment.

![Figure 8. Experiment with the NRS and one restricted/dangerous area: positions and cumulative reward over time steps.](image)

The robot navigates initially to the resting position (7). Once the person is seen moving to east and is somewhat near 1, the robot starts moving to the restricted/dangerous cell. The person reaches that area before the robot since real-time navigation takes more time than a person’s movement. Despite this, the robot does not take more than a few seconds to catch up with the person at 1. The robot informs the person that they should not be there. The person leaves the area and later the environment. Finally, the robot returns to 7.

C. Experiment 1 with $r = 2$

The following two experiments consider two restricted places, in cell 1 and 12. This trial explores a situation where a person moves to a restricted/dangerous place and does not leave for some time after being told to avoid that area.

Results of the experiment can be visualized in Figure 9 and Table VII.

![Figure 9. Experiment with two restricted/dangerous areas: positions and cumulative reward over time steps.](image)

Initially, the robot moves the equidistant cell (7) from both restricted/dangerous areas. Once the person is seen near 12, the robot begins to move to 12. Since the person never leaves a region near the restricted/dangerous cell, the robot never stops moving to 12. Once the robot is at 12, the person is informed that they not stay there. The person does not respect

1https://www.youtube.com/watch?v=slgOpOFrIvc
2https://www.youtube.com/watch?v=2S4e5QvYDNM
3https://www.youtube.com/watch?v=PrzhCS99Mhw
Once the person moves to the other restricted/dangerous area (12), the robot also goes to that place.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
<th>Obs Robot</th>
<th>Obs Person</th>
<th>Obs Velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>East</td>
<td>7</td>
<td>No person</td>
<td>Not moving</td>
</tr>
<tr>
<td>2</td>
<td>Do nothing</td>
<td>7</td>
<td>7</td>
<td>North higher</td>
</tr>
<tr>
<td>3</td>
<td>Do nothing</td>
<td>7</td>
<td>6</td>
<td>East 0-0.2</td>
</tr>
<tr>
<td>4</td>
<td>East</td>
<td>5</td>
<td>1</td>
<td>Not moving</td>
</tr>
<tr>
<td>5</td>
<td>East</td>
<td>3</td>
<td>4</td>
<td>East 0-0.2</td>
</tr>
<tr>
<td>6</td>
<td>East</td>
<td>1</td>
<td>2</td>
<td>Not moving</td>
</tr>
<tr>
<td>7</td>
<td>Do nothing</td>
<td>1</td>
<td>4</td>
<td>Not moving</td>
</tr>
<tr>
<td>8</td>
<td>Do nothing</td>
<td>1</td>
<td>8</td>
<td>Not moving</td>
</tr>
<tr>
<td>9</td>
<td>Do nothing</td>
<td>1</td>
<td>6</td>
<td>West 0-0.2</td>
</tr>
<tr>
<td>10</td>
<td>West</td>
<td>3</td>
<td>8</td>
<td>Not moving</td>
</tr>
<tr>
<td>11</td>
<td>West</td>
<td>5</td>
<td>6</td>
<td>Not moving</td>
</tr>
<tr>
<td>12</td>
<td>West</td>
<td>7</td>
<td>8</td>
<td>South 0.2-0.7</td>
</tr>
<tr>
<td>13</td>
<td>Do nothing</td>
<td>7</td>
<td>7</td>
<td>Not moving</td>
</tr>
<tr>
<td>14</td>
<td>Do nothing</td>
<td>7</td>
<td>10</td>
<td>North 0.2-0.7</td>
</tr>
<tr>
<td>15</td>
<td>North</td>
<td>8</td>
<td>11</td>
<td>Not moving</td>
</tr>
<tr>
<td>16</td>
<td>West</td>
<td>10</td>
<td>11</td>
<td>East 0-0.2</td>
</tr>
<tr>
<td>17</td>
<td>West</td>
<td>12</td>
<td>11</td>
<td>Not moving</td>
</tr>
</tbody>
</table>

Table VIII
EXPERIMENT 2 WITH TWO RESTRICTED/DANGEROUS AREAS: POLICIES AND OBSERVATIONS OVER TIME STEPS.

E. Analysis

Results evidenced that the model can be used to accomplish the proposed task. The usage of a POMDP in this task is an advantage since the policies are chosen according to the belief states whose values evolve over time. Policies are only selected when there is a high degree of confidence of the person’s position and movement direction. The reward models also prove to be useful as the robot always took the shortest path between areas of interest and executed actions that resulted in penalties but proved better in the long run, thus maximizing the agents cumulative reward. Every time the robot reached a restricted/dangerous cell, the agent received a high positive reward. This compensated the navigation costs and the penalties of being outside of the equidistant cell.

When the robot was at the restricted/dangerous place, it only left once the person was seen outside of it for two consecutive times. Results show that two belief updates are necessary to increase the degree of certainty that the person is no longer at the spot. This is considered to be a positive outcome since it means that two potentially negative situations where the robot would be leaving too soon can be discarded: a false positive placing the person in another position; and the person leaving and immediately coming back. The fact that it takes two observations for the robot to leave is something that would require the usage of memory, should a heuristic be used instead of a POMDP.

In the simulation an assumption that the action and observations are instant was made which is the most advisable measure to test the model. A person is observed continuously moving in the same direction and transitioning between cells at the same pace as the robot. The robot only initiates movement once there is a high probability of the person moving to east. In simulation, the robot and the person reach the restricted area at the same time step. Thus, the person is advised not to be there immediately at their arrival. The limitations of using an actual
robot are clear. In experiments with one restricted area, the person exhibits the same type of behaviour. However, when using the robot, the person reached the restricted/dangerous place seconds before the robot. This means that while the model can be used for this task, there will always be a delay between the time person and robot reach an area of interest. Despite the delay, the robot still acted according to the observations perceived and the task was completed.

Experiments with two restricted/dangerous zones showed that the robot will not move to one of the zones unless the person does so. While the person is between both areas, the robot will stay in the equidistant spot. The moving direction of a person influences their position in the next step. Results of the experiments showed that if the observation received is not consistent with the previous moving direction, the belief that the person is in the state observed will not be that high.

VII. CONCLUSIONS AND FUTURE WORK

The work carried out is considered positive. It was possible to perform optimal planning and apply it to interact with a person in an indoor space using a NRS. The model developed proved that the POMDP framework is useful in task planning. The model would have to be adjusted to other environments but the fact that it could be successfully implemented in the test environment shows that it could have the same result in other indoor spaces. While this task may be applied in the conditions of the experimental work of this thesis, it is better suited for larger open spaces.

Navigation between waypoints in cells took some time due to both the robot’s physical limitations and the existing navigation system. When the robot moves, it needs to update a costmap, containing obstacles, of the environment. This prevented immediate navigation when a goal was received. Applying this model in conditions where these limitations do not exist would be useful.

The POMDP model took into account one person in the environment. In a large open space with multiple people, the model would have to consider extensive state and observation spaces. This would increase the computation time of the model exponentially. One possible solution could be having multiple robots in the environment. Each robot would be responsible for a specific region or particular people, thus the model would be divided into multiple sub-models. Another solution could be decreasing the number of possible positions for a person and only consider if the person is in, near or far from a restricted/dangerous area.

The movement direction implemented in the model took advantage of the fact that a person would necessarily move to east or west to reach a dangerous place. In a larger environment, that information would not be enough. It would be necessary to use further directions, possibly using orientations with an associated angle.

Planning occurred offline because the environment was known. If that was not the case, one elegant solution of adapting to unplanned occurrences would be online planning which allows to change the number of variables and existing models according to each situation. In the future, it would be interesting to study this model in online planning.

Machines processing power have been increasing over time and it is safe to assume that current complex POMDP models will be solvable in a shorter period of time in the future. It is also pertinent to assume that NRSs will take less time executing actions and sensing their surroundings in the future. For that reason, there is a lot of potential of this task being applied in multiple scenarios and of POMDPs becoming of regular use in human-aware planning.

REFERENCES