Adaptive Robot Navigation through Integrated Task and Motion Planning

Phani Teja Singamaneni[1], Alessandro Umbrico[2], Andrea Orlandini[2] and Rachid Alami[1,3]
[1] LAAS-CNRS
Toulouse, France
[2] CNR – Institute of Cognitive Sciences and Technologies (ISTC-CNR)
Rome, Italy
[3]ANITI, Université de Toulouse
Toulouse, France

Abstract

Robots acting in real-world environments usually interact with humans. Interactions may occur at different levels of abstraction (e.g., process, task, physical), entailing different research challenges (e.g., task allocation, human-robot joint actions, robot navigation). When acting in social situations, robots should recognize the context and behave in different manners, so as to act and interact in a correct and acceptable way. We propose the integration of task and motion planning to contextualize robot behaviors for social navigation. The main idea is to leverage the contextual knowledge of a deliberative task planner to dynamically adapt the navigation behaviors of a robot and enhance human-robot interaction. More specifically, we propose a holistic model of tasks and human features and a mapping from task-level knowledge to motion-level knowledge to constrain the generation of robot trajectories. The proposed framework is tested in simulation for some commonly occurring scenarios in a hospital.

Introduction

Robots acting in situations requiring direct or indirect interactions with humans should realize behaviors that take into account also a social dimension. A set of implemented behaviors should not be only technically valid and efficient but also acceptable by humans (Rossi, Ferland, and Tapus 2017). There is a crucial need of considering a social perspective in order to *meet* the expectations of humans in different (social) contexts and, realize behaviors that are safe, reliable, and legible. In Human-Robot Interaction (HRI), it is particularly important to reason about *how* tasks are carried out by a robot in order to do the *right action* in the *right way* and comply with the so-called *social norms* (Triebel et al. 2016; Bruno et al. 2019; Awaad, Kraetzschmar, and Hertzberg 2015). Nevertheless, human behaviors are usually only partially predictable. From a control perspective, the presence of humans constitutes a source of *uncertainty* concerning, e.g., their goals, beliefs, and intentions (Clodic and Alami 2021; Clodic et al. 2017). This uncertainty raises robot control issues and strongly affects the way a robot achieves its goals.

To deploy more natural and acceptable behaviors, robots need advanced *reasoning capabilities* and intelligent controllers that take into account: *who* is the human a robot interacts with; *what* are the objectives of the interactions; *how* to achieve them; *when* to execute the needed actions, and; *where* interactions take place. In general, it is necessary to reason about both technical and social aspects of the interactions in order to, respectively, realize correct behaviors and adapt robot behaviors to different contexts and human users.

Implementing "intelligent behaviors" requires investigating several research directions that lead to the integration of Robotics and Artificial Intelligence (AI) (Lemaignan et al. 2017; Ingrand and Ghallab 2017). General interaction capabilities of robotic platforms should be *customized* according to the specific features of a scenario as well as the preferences and needs of users (Cortellessa et al. 2021; Moro, Nejat, and Mihailidis 2018). In this regard, it is paramount to endow the robot with *contextual knowledge* about human users, social environments, and (social) tasks to be performed. On the one hand, such knowledge allows robots to *personalize* their general interactions capabilities (i.e., *behaviors*) to the specific needs of a user. On the other hand, it provides a means for robots to *adapt* their behavior execution over time according to the changing or evolving states of users (Umbrico et al. 2020).

In this work, we propose a novel integrated Task And Motion Planning (TAMP) approach to enhance the awareness of the social navigation skills of robots. This approach relies on a motion planning framework, called CoHAN (Singamaneni, Favier, and Alami 2021; 2022), which allows the tuning of human-aware navigation behaviors. It exposes a number of motion parameters that are used by a task planner, called PLATINUm (Umbrico et al. 2017) to dynamically adapt motion behaviors to the expected social context. To this aim, the paper proposes a *holistic model* characterizing domain and task requirements as well as human features at both domain and geometric levels. The feasibility of the approach is evaluated in an assistive scenario where a robot is requested to dynamically change motion behaviors according to different types of tasks, environmental features as well as interacting features of involved humans.

Human-Aware Task and Motion Planning

Endowing a control system with a well-structured model of humans and social contexts is crucial to synthesize flexible and effective robot behaviors. Indeed, there are several human and social-related variables that can affect motions

and *interaction styles* of a robot in a certain social context. Usually, works in social navigation mainly focus on the geometric aspects of the implemented motions (Kruse et al. 2013). Humans are generally considered as dynamic obstacles whose geometric model is defined by taking into account different aspects e.g., proxemics (Ferrer et al. 2017; Rios-Martinez, Spalanzani, and Laugier 2015; Mead and Matarić 2017) or emotional states (Cavallo et al. 2018). There are additional factors concerning human intention, perspectives, or social norms that should be considered to reliably plan robot motions (Che, Okamura, and Sadigh 2020; Repiso, Garrell, and Sanfeliu 2022; Beraldo et al. 2022). Furthermore, works usually focus on single navigation/interaction episodes ignoring more abstract knowledge about the interaction skills of involved humans, sequences of motions (i.e., robot plans), and (domain-level) "motivations" that lead the robot to act in a (social) situation.

Depending on the specific requirements of the domain-level task being performed and the *qualities* of involved humans, the execution of needed (social) navigation skills would consider different priorities, safety requirements, and different performance constraints. Such contextual knowledge impacts the *interaction style* of a robot and the way navigation behaviors are actually implemented. In this regard, we propose a *holistic model* for social navigation tasks to characterize interacting behaviors from different (synergetic) perspectives: (i) the *domain perspective* considers technical and performance aspects of a task being executed; (ii) the *human perspective* considers the interacting skills, qualitative features, and preferences of humans involved in the execution of a task; (iii) the *robot perspective* considers the acting skills and execution modalities of trajectories; (iv) the *environment perspective* considers social features of the physical space where the execution of a task takes place. Each of these perspectives contributes to different levels of abstraction while defining context-aware robot behaviors. We integrate this holistic model into a novel Task and Motion planning (TAMP) approach to synthesize flexible robot behaviors.

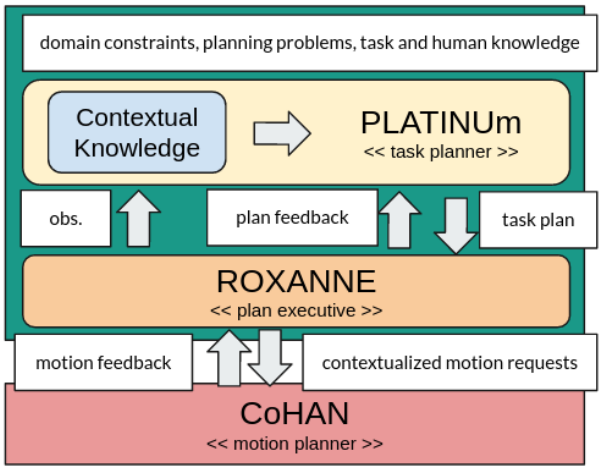

Figure 1: Integrated Task and Motion planning architecture.

Fig. 1 shows the designed architecture and the resulting control flow. The task planner reasons at a high level of abstraction taking into account the domain requirements, the functional capabilities of the robot, and the interaction skills of humans. It deals with task decomposition, task assignment, and temporal sequencing of needed actions. We rely on the timeline-based planner PLATINUm[1] and the executive framework ROXANNE[2] (Cialdea Mayer, Orlandini, and Umbrico 2016; Umbrico et al. 2017). The motion planner reasons at a lower level of abstraction taking into account the geometrical features of the environment, the robot, and the involved humans. We rely on the human-aware motion planning framework CoHAN[3] which exposes a number of parameters to adapt the planning of motion trajectories to different contexts (Singamaneni, Favier, and Alami 2021; 2022).

The idea behind the proposed TAMP approach is to leverage the domain-level knowledge of the task planner to enrich the motion planner with *contextual knowledge* about tasks and involved humans. It is important to point out that a model of humans is necessary at both task and motion planning levels. Humans indeed should be characterized from both a functional level (necessary for the task planner) and a geometric level (necessary for the motion planner). The TAMP approach integrates such knowledge to support human-aware reasoning.

Task-level Knowledge

This section describes the task planning knowledge characterizing the requirements, motivations, and objectives that influence the robot's actions. Table 1 shows the variables considered at this level of abstraction: (i) the *environmental context* in which a task is performed (e.g., crowded spaces, public or private environments); (ii) the *risk* of a task with respect to the safety of humans and; (iii) the *performance* of a task (e.g., flexible tasks whose execution does not require rigid adherence to the nominal duration or strict tasks whose execution cannot be delayed).

The higher the (cumulative) score, the lower the need of considering human-related constraints. Depending on the cumulative score of the variables, we define three classes of tasks: (i) **Technical-critical tasks** (score $\in (6, 9]$) focus on technical requirements mainly. The motion planner would thus relax social constraints in favor of optimized and efficient motions; (ii) **Interaction-critical tasks** (score $\in (3, 6]$) require a trade-off between technical and social constraints when planning motions; (iii) **Social-critical tasks** (score $\in [0, 3]$) focus on social requirements. The motion planner would almost ignore optimal trajectory to favor safe and acceptable motions.

Table 2 shows the set of variables characterizing the interaction skills of humans. These variables are a subset of the *International Classification of Functioning, Disability, and*

[1] https://github.com/pstlab/PLATINUm.git

[2] https://github.com/pstlab/roxanne_rosjava.git

[3] https://github.com/sphanit/CoHAN_Planner.git

Table 1: Variables characterizing domain-level knowledge of a task.

Parameter	Value Set	Value Range	Description
Social Context	{crowded, public, private, robotic}	[0, 3]	Describe the environmental context in which a task is supposed to be performed. Higher values correspond to a lower predominance of humans, and consequently higher availability of space to plan robot motions.
Risk	{critical, high, average, low}	[0, 3]	Describe the risk of the execution of a task with respect to the safety of humans. Tasks with low risk, for example, would allow the execution of optimal trajectories that are not necessarily social. Vice versa, tasks with high risk would imply the execution of social (and non-optimal) trajectories.
Performance	{none, flexible, regular, strict}	[0, 3]	Describe the required level of performance during the execution of the motions. Higher values imply stricter adherence to performance optimization when planning robot motions.

Health (ICF)[4] proposed by the World Health Organization (WHO). The ICF framework describes the level of functioning of a person and has been proposed in robotics to adapt the interactions (Umbrico et al. 2020; Kostavelis et al. 2019; García-Betances et al. 2016). Each variable is associated with an integer value[5] expressing the level of functioning of a person with respect to a particular aspect.

The rationale behind the use of this subset of ICF is to estimate *human uncertainty* with respect to the interaction with the robot. The higher the cumulative score of the variables the higher the uncertainty (i.e., higher impairments of interaction-related aspects). Depending on the cumulative scores of the variables we define three classes of humans: (i) **Fragile** humans (score $\in (25, 44]$) have limited interaction skills (e.g., low hearing or seeing functioning) and unstable motions (e.g., unstable walking, equilibrium issues, or low attention). This category entails conservative/prudent robot motions since no assumption can be made on the actual state or motions of the human (maximum uncertainty); (ii) **Average** humans (score $\in (13, 25]$) have minimal interaction skills and sufficiently stable motions. This category allows the robot to make some assumptions about the expected behaviors of the interacting humans and thus perform some level of optimization and planning of motions (average uncertainty); (iii) **Reliable** humans (score $\in [0, 13]$) can reliably interact with robots and perform mutual adaptation to robot motions. This category allows a robot to achieve a higher level of optimization since the behavior of the human is predictable to some extent (minimum uncertainty).

This knowledge with the discussed classes of tasks and humans is used within the task planning model to contextualize the interacting tasks of the robot. The task planner would thus provide the motion planner with contextual information useful to dynamically adapt the planning of motion trajectories.

Motion-level Knowledge

The framework CoHAN generates flexible motion trajectories by taking into account observed intentions of humans and supporting perspective-taking (Singamaneni, Favier, and Alami 2021; 2022). CoHAN exposes a number of navigation parameters that can be used by a task planner to support a finer tuning of motion trajectories. Table 3 shows the sets of motion parameters exposed by CoHAN and used within the proposed TAMP approach to dynamically constrain the generation of robot trajectories.

The parameter variables can be grouped into three sets. The first set characterizes the qualities of robot motions. In addition to variables limiting the speed and acceleration of the robot, the variable *plan* specifies the "look ahead" of the motion planner, determining the "length" of the trajectories. The variable *band* specifies the level of collaboration of the robot in solving motion conflicts with humans (e.g., deadlocks in narrow passages). The second set characterizes the qualities of human motions. In addition to variables estimating the velocity and acceleration of the human, the variable *radius* specifies the volume around the human body thus determining the proxemics constraints for the motions of the robot. The variable *field of vision* estimates the breadth of the field of vision of the human, determining whether the robot is visible or not to the human. The third set characterizes the social qualities of robot motions. The variable *safety* specifies the level of safe distance the robot must maintain while moving. The variable *visibility* determines the way the robot should enter the field of vision of the human from behind. The variable *hidden human* allows the robot to be cautious about occluded regions from where the human might emerge (Singamaneni, Favier, and Alami 2022). All these variables are meant to realize behaviors that are acceptable to humans.

Combining Perspectives

To dynamically set the values of the motion parameters according to the expected interaction context, it is necessary to combine the domain-level knowledge of the task planner with the geometry-level knowledge of the motion planner. Depending on the classification of the task and the involved human, the task planner would determine *patterns* of motion parameter values.

For example, *social-critical tasks* requiring the interaction between a robot and a *fragile user* would entail a prudent navigation behavior of the robot. In such a case, the task planner would dispatch navigation requests by setting the variable *Radius* to "big" (i.e., avoid the robot moving too close to the human); the variable *Planning horizon* to "max" (i.e., plan long trajectories in order to be more adaptive when approaching the human); the variable *Field of vision* to "narrow" (i.e., assume the user would see the robot only when in front of him/her); the variable *Hidden humans* to "max" (i.e., let the robot be very prudent when entering rooms or turning

[4] https://icd.who.int/dev11/l-icf/en

[5] ICF scores are defined within a 5-point Likert scale: (i) 0 means no impairment; (ii) 1 means soft impairment; (iii) 2 means medium impairment; (iv) 3 means serious impairment; (v) 4 means full impairment.

Table 2: Variables characterizing human-level knowledge of a task.

ICF Area	ICF variable	Value Range	Description
Mental Functioning	Attention	[0, 4]	Specific mental functions of focusing on an external stimulus or internal experience for the required period of time.
	Memory	[0, 4]	Specific mental functions of registering and storing information and retrieving it as needed.
	Orientation	[0, 4]	General mental functions of knowing and ascertaining one's relation to time to place, to self, to others, to objects, and to space.
	Perception	[0, 4]	Specific mental functions of recognizing and interpreting sensory stimuli.
Sensory	Hearing	[0, 4]	Sensory functions relating to sensing the presence of sounds and discriminating the location, pitch, loudness, and quality of sounds.
	Seeing	[0, 4]	Sensory functions relating to sensing the presence of light and sensing the form, size, shape, and color of the visual stimuli.
	Vision	[0, 4]	Mental functions involved in discriminating shape, size, color, and other ocular stimuli.
Mobility	Body Position	[0, 4]	Staying in the same body position as required, such as remaining seated or remaining standing for carrying out a task, in play, work, or school.
	Movement Control	[0, 4]	Functions associated with control over and coordination of voluntary movements.
	Muscle Tone	[0, 4]	Functions related to the tension present in the resting muscles and the resistance offered when trying to move the muscles passively.
	Walking	[0, 4]	Moving along a surface on foot, step by step, so that one foot is always on the ground, such as when strolling, sauntering, walking forwards, backward, or sideways.

Table 3: Motion variables exposed by CoHAN.

Type	Parameter	Value Set	Description
Robot	Velocity	{min, nominal, max}	Set the velocity limits of the implemented motions of the robot.
	Angular velocity	{min, nominal, max}	Set the angular velocity of the implemented motions of the robot.
	Acceleration	{min, nominal, max}	Limit the maximum acceleration of the motions of the robot.
	Planning horizon	{min, normal, max}	Set the "look ahead" of the planned motion trajectories of the robot.
	Band tightness	{loose, medium, tight}	Set the collaborative level of the implemented behavior of the robot.
Human	Radius	{small, medium, big}	Estimate the volume of the human determining the *proxemics constraints* for the motion of the robots.
	Velocity	{min, nominal, max}	Estimate the speed of observed human motions over a certain cartesian direction.
	Angular velocity	{min, nominal, max}	Estimate the angular speed of observed human motions.
	Field of vision	{narrow, normal, wide}	Estimate the breadth of the field of vision of a human and thus the "eye contact" with the robot.
	Band tightness	{loose, medium, tight}	Estimate the possibility of a human changing his/her path.
Social	Safety	{none, min, nom, max}	This variable specifies the level of safety a robot must support while moving.
	Relative velocity	{none, min, nom, max}	This variable reduces the velocity of a robot as its distance from humans decreases and allows the robot to quickly change the path (moves to one side).
	Visibility	{none, min, nom, max}	This variable allows a robot to avoid entering the human's field of view very closely from behind.
	Hidden humans	{none, min, nom, max}	This variable makes the robot cautious about the occluded regions from where a human might emerge.

corners in corridors); etc.

Table 4 maps the three classes of tasks to the motion variables characterizing the navigation skills of the robot and Table 5 maps the three classes of humans to the motion variables characterizing the behavioral model of the human. Finally, Table 6 maps combinations of task/human classes to the social-related motion variables.

Table 4: Map Task classes to Robot-related motion parameters.

Task Class	Velocity	Angular Velocity	Acceleration	Planning Horizon	Band Tightness
Technical	max	max	max	min	tight
Interaction	nominal	nominal	nominal	nominal	medium
Social	min	min	min	max	loose

Experimental Evaluation

To assess the proposed TAMP approach we consider an in-hospital scenario where a socially interacting robot is deployed to support patients and healthcare personnel. The domain entails a variety of *social situations* e.g., approaching

Table 5: Map Human classes to Human-related motion parameters.

Human Class	Radius	Velocity	Angular Velocity	Field of Vision	Band Tightness
Fragile	big	min	min	narrow	tight
Average	medium	nominal	nominal	nominal	medium
Reliable	small	max	max	wide	loose

fragile users, navigating inside crowded corridors, or entering rooms populated by users whose view is occluded to the robot. In this context, the robot should perform different types of tasks (e.g., drug delivery, patient monitoring, technical support to healthcare professionals) and interact with different categories of humans (e.g., fragile patients and more reliable healthcare professionals) within different environments (e.g., rooms, corridors).

The integrated approach has been tested in simulation using stage_ros package[6]. Fig. 2 shows the part of the hospital map used for all the scenarios with humans' (circles) and the robot's (square) positions. There are a total of six hu-

[6] http://wiki.ros.org/stage_ros

Table 6: Map combinations of human and task classes to Social-related motion parameters.

Classes	Safety	Relative Velocity	Visibility	Hidden Humans
Technical+Fragile	max	nominal	min	nominal
Technical+Average	min	min	nominal	nominal
Technical+Reliable	min	min	min	min
Interaction+Fragile	max	nominal	nominal	max
Interaction+Average	nominal	nominal	nominal	nominal
Interaction+Reliable	min	min	nominal	nominal
Social+Fragile	max	max	max	max
Social+Average	nominal	nominal	max	max
Social+Reliable	min	nominal	nominal	nominal

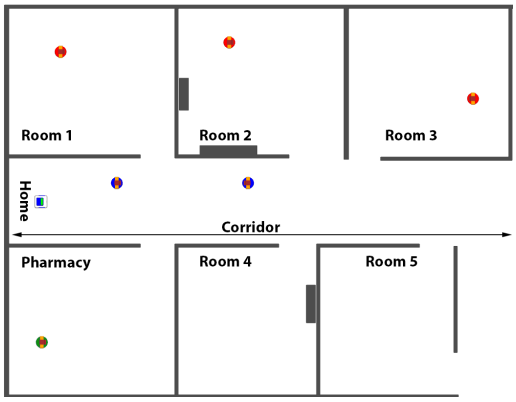

Figure 2: Hospital scenario simulated in stage_ros

mans in the setting with three humans representing patients in three different rooms, one pharmacist, and two others acting as pedestrians in a corridor. The humans in red circles are the patients, the ones in blue are the pedestrians and the human in green is the pharmacist. In this environment, we have designed and tested three scenarios: (i) A *drug delivery scenario* requires the robot to reach the pharmacy to pick up some drugs and deliver them to a particular patient located in a known room; (ii) A *patrolling scenario* requires the robot to move inside the different rooms of the floor to monitor the general health conditions of patients; (iii) An *emergency scenario* requires the robot to quickly reach the room hosting the patient asking for help. In all of these scenarios, humans are static in the rooms and the pharmacy, whereas they are dynamic from time to time in the corridor.

The central aspect is the need for dynamically adapting motion behaviors according to the social situations characterizing the execution of each task (e.g., navigating the corridor, entering a room with patients, approaching patients or healthcare professionals). In particular, it is necessary to adapt motion behaviors within the execution of the same domain-level task.

Task Planning Model

The task planning model of the scenarios has been defined following the timeline-based formalism (Cialdea Mayer, Orlandini, and Umbrico 2016) [7]. Broadly speaking, the model consists of a number of *state variables* characterizing temporal behaviors of domain features, and a number of *synchronization rules* specifying global constraints coordinating the temporal evolution of single state variables. While state variables specify local constraints characterizing the correct dynamics of the modeled domain features, synchronization rules specify global constraints coordinating state variables to achieve complex goals (e.g., perform a high-level domain task).

We have defined four state variables whose values are predicates asserting states or activities the associated features may respectively assume or execute over time. The state variable *RobotBase* models the end-effector allowing the robot to (autonomously) navigate the environment. The value *At(?l)* specifies the robot is still in a specific location (i.e., the parameter *location* of the predicate). The value *MoveTo(?l, ?t, ?u)* models the generic action of moving the robot towards a particular goal location (?l). The other parameters of the predicate contextualize the motion with respect to the class of task (?t) the robot is supposed to perform and the class of human (?u) the robot is expected to interact with.

The state variable *RobotMotionController* contextualizes navigation skills with respect to the structure of the environment. The values *Still(?l)* and *NavigateTo(?l, ?t, ?u)* characterize the motion of the robot outside the rooms. The other values e.g., *Enter(?l, ?t, ?u)*, *Inside(?l)*, *Approach(?l, ?t, ?u)*, etc. characterize the motion of the robot inside the rooms. The state variable *RobotSkill* describes the interacting skills of the robot. The values represent complex actions (e.g., *PickDrug(?l1, ?l2, ?l3)*, *DeliverDrug(?l1, ?l2, ?l3)*) the robot can perform through its end-effectors. The state variable *RobotService* describes the high-level tasks a robot should perform within the domain. Each value represents a planning goal and is associated with a particular scenario (i.e., *DeliverDrug(?l1, ?l2, ?l3)*, *Patrolling(?l1, ?l2, ?l3)*, and *Emergency(?l1, ?l2, ?l3)*).

State variables have a simple structure with a single value as a stable state (e.g., the values *Idle()*, *Still(?l)* and *At(?l)*) and other values as actions with bounded duration (e.g., *DeliverDrug(?l1, ?l2, ?l3)*, *PickDrug(?l1, ?l2, ?l3)*, *HelpPatient(?l1, ?l2, ?l3)*). Value transitions thus go from a stable state value (e.g., *Idle()*) to action values (e.g., *DeliverDrug(?l1, ?l2, ?l3)*) and vice versa. The state variable *RobotMotionController* has a more complex structure depicted in Fig. 3. It distinguishes between the robot moving outside rooms (i.e., *NavigateTo(?l, ?t, ?u)*) and inside rooms (e.g., *Enter(?l, ?t, ?u)*, *Approach(?l, ?t, ?u)*) supporting a finer tuning of the motion parameters.

Synchronization rules then organize state variables in a hierarchical fashion and specify the decomposition of high-level tasks (i.e., values belonging to more abstract state variables) into incrementally simpler tasks (i.e., values belong-

[7] Due to space limitations it was not possible to add an overview of the planning formalism. We invite readers to refer to (Cialdea Mayer, Orlandini, and Umbrico 2016) for an exhaustive description of the timeline-based formalism.

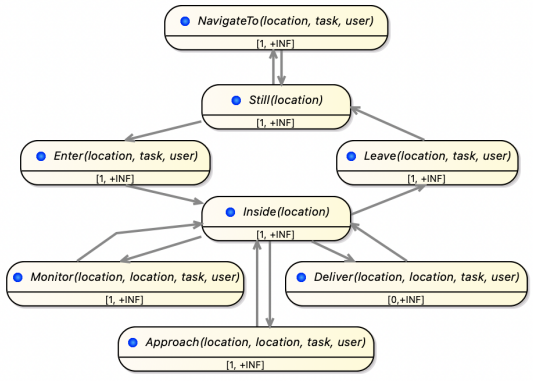

Figure 3: State variable *RobotMotionController*.

ing to lower-level state variables). For example, the high-level task *DeliverDrug(?l1, ?l2, ?l3)* is first decomposed into values of the state variable *RobotSkill* denoting complex actions necessary to accomplish the task correctly (e.g., *PickDrug(?l1, ?l2, ?l3)*). Each complex action is then further decomposed into a number of contextualized navigation skills (e.g., *NavigateTo(?l, ?t, ?u)*). Rules may specify temporal relations constraining the correct execution of the sub-tasks. For example, within the high-level task *DeliverDrug(?l1, ?l2, ?l3)* of the *RobotService* state variable, the *BEFORE* temporal relation (Allen 1983) specifies the correct sequencing of the complex actions *PickDrug(?l1, ?l2, ?l3)*, *DeliverDrug(?l1, ?l2, ?l3)* and *GoHome()* of the *RobotSkill* state variable. Each navigation skill is then further decomposed into the primitive motion actions (i.e., instances of *MoveTo(?l, ?t, ?u)*) that are dispatched to the motion planner for execution [8].

Given the current focus on human-robot interaction, the planning model does not consider battery constraints. However, timeline-based formalism supports different types of resources (Umbrico et al. 2018) and the model can be easily extended to take into account robot autonomy (i.e., battery consumption/production).

ROS-based Implementation and Integration

The integrated task and motion planning approach depicted in Fig. 1 has been implemented in ROS Melodic using: (i) PLATINUm (Umbrico et al. 2017) as a timeline-based task planning engine; (ii) ROXANNE as ROS-compliant executive for timeline-based plans, and; (iii) CoHAN (Singamaneni, Favier, and Alami 2021) as a motion planner implementing the navigation skills of the robot.

ROXANNE is a ROS package supporting the development of goal-oriented plan-based controllers. It encapsulates PLATINUm as a timeline-based planning engine and provides a ROS-compliant executive. Interactions with ROXANNE and the underlying task planning system are realized through a set of topics exchanging custom ROS messages: (i) receiving planning requests (i.e., *ActingGoal*); (ii) dispatching plan tokens as execution requests for a robot (i.e., *TokenExecution*), and; (iii) receiving related execution feedback (i.e., *TokenExecutionFeedback*).

The actual set of topics used by ROXANNE can be set through a dedicated configuration file. In this case, we configure ROXANNE with a single goal topic and a pair of dispatching and feedback topics for the execution of navigation skills. The dispatching topic allows PLATINUm to send CoHAN contextualized motion execution requests. The content of a dispatched message (*TokenExecution*) is a token composing the timeline of *RobotBaseType*, instantiating the value *MoveTo(?l, ?t, ?u)*. The feedback topic allows PLATINUm to receive information from CoHAN about the actual execution of dispatched commands. The content of a feedback message (i.e., *TokenExecutionFeedback*) consists of a *code* denoting the result of the execution where a value of 0 means successful execution and values > 0 are used to denote different types of failures.

The control flow of the robot is initiated by the task planner when receiving a planning request through the goal topic of ROXANNE. A goal request instantiates one of the values of the state variable *RobotService* (e.g., the value *DeliverDrug(?l1, ?l2, ?l3)*) thus representing a high-level planning task to perform. The task planner synthesizes a set of valid timelines for each of the state variables of the model, following the decomposition encoded by the synchronization rules. The obtained timeline-based plan is then executed through ROXANNE.

Given a planned instance of the value *MoveTo(?l, ?t, ?u)* (i.e., a token of the timeline of the state variable *RobotBase*), ROXANNE encapsulates related information into a *TokenExecution* message and sends it through the dispatching topic. A simple ROS package is developed that receives this request and dynamically maps the parameters according to the values of *?task* and *?user*, following the mappings in Tables 4, 5, and 6. It then updates the motion parameters of CoHAN before calling the motion planning service. When the motion execution is complete, CoHAN sends a *TokenExecutionFeedback* to ROXANNE notifying the result. For each dispatched request, ROXANNE waits for feedback before executing the next tokens of the timeline.

Results and Discussion

The variety of social situations occurring in the designed assistive domain is well-suited to assess the contextual and human-aware navigation capabilities of the proposed TAMP approach. To this aim, the evaluation analyzes and compares the results of two configurations: (i) *cohan* showing the behavior without the use of contextual information from the task planner; (iii) *cohan+platinum* showing the behavior of the proposed TAMP approach. We now proceed to the description of the parameters in each scenario and analyze the results.

Patrolling Scenario The goal in this scenario is to navigate to each room with a patient and monitor them taking into consideration their fragile state. The list of parameters for this scenario is shown in Table 7. The robot goes to each room to monitor the patient and the complete task consists of a total of 13 navigational phases (or steps). The veloc-

[8]The planning model is available on the GitHub repository of ROXANNE - https://github.com/pstlab/roxanne_rosjava/blob/master/domains/assistive_map1_v0.1.ddl

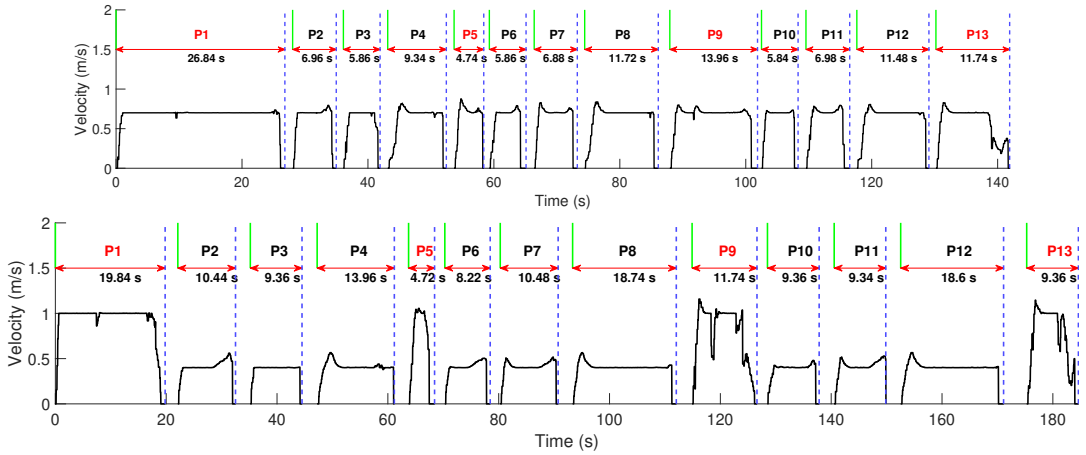

Figure 4: Velocity profiles and times for navigation phases (red: corridor ones) in the patrolling scenario. Top: *cohan*, Bottom: *cohan+platinum*

Navigate corridor	Task: Technical, Human: Average
Enter patient's room	Task: Social, Human: Average
Monitor patient	Task: Social, Human: Fragile
Leave room	Task: Social, Human: Average

Table 7: Parameters for patrolling and monitoring

ity profiles of the robot and the time taken in each navigation phase are presented in Fig. 4. The top part of the figure shows the run in *cohan* configuration, while the one at the bottom shows the run in *cohan+platinum* configuration. The green line represents the start and the blue dashed line represents the end of each navigation phase. The subsequent figures follow the same conventions.

P1, P5, P9, and P13 are phases of navigating the corridor, and the ones in between consist of entering, monitoring, and leaving each room. From the top part of Fig. 4 showing the *cohan* setting, it can be seen that the robot moves with a maximum velocity of around $0.7\,\mathrm{m/s}$. However, this may not be ideal for monitoring patients who can have limited reactive capabilities. Hence, the robot should be more careful (and slower) around the patients which can be achieved by updating the parameters (*cohan+platinum*) as shown in the bottom part of Fig. 4. Since the corridor navigation phases are simply technical, the robot could move with a larger velocity and use minimal human-aware capabilities for navigation. It makes the robot pass through these phases faster (comparing the times) when compared to the *cohan* setting. It might also affect the robot's trajectory as shown in Fig. 5. The color of the trajectory changes from blue at the start to red towards the end. Having toned down human-aware capabilities makes the robot react to the human only when required (Fig. 5 (b)) instead of adapting its trajectory early as in the case of Fig. 5 (a). This makes the robot traverse the corridor faster by reducing the effect of humans on the trajectory.

The total time taken for the robot to complete the task with dynamic parameter adaptation is $184.64\,\mathrm{s}$ and with constant parameters, it is $141.94\,\mathrm{s}$. Even though there is an overhead of $42.7\,\mathrm{s}$, the integrated approach moves the robot

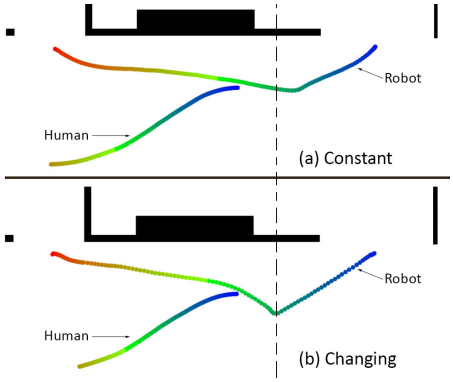

Figure 5: Trajectories in Phase 9 of patrolling scenario.

more safely around the fragile humans in the environment. Moreover, in a human-aware planning setting, the fastest approach may not be the ideal approach. From Fig. 4, it can be seen that there is a small time gap between the end of one phase and the start of another. This timing gap is attributed to the communication delay associated with the task planner and the motion planner, and the time to update the parameters of CoHAN. Considering only the times of the navigation phases, the total execution times for *cohan* and *cohan+platinum* settings are $128.20\,\mathrm{s}$ and $154.16\,\mathrm{s}$. This gives us an overhead time of $2.34\,\mathrm{s}$ per phase in the dynamic setting and $1.06\,\mathrm{s}$ per phase in the constant setting. Although these overhead times are not large for real-world applications, we plan to reduce them in the future version of TAMP.

Emergency Scenario In this scenario, the robot has to rush to the patient in a room to assist in an emergency. For the setting, the list of parameters is updated as shown in Table. 8. Unlike the previous scenario, the robot moves at a larger speed to rush to the patient in the *cohan+platinum* setting compared to the *cohan* setting. From Fig. 6, it can be seen that each phase in the *cohan+platinum* configuration is executed in less or almost the same time as the *cohan* setting. The total time for *cohan+platinum* setting is $70.80\,\mathrm{s}$ with an overhead time of $2.24\,\mathrm{s}$ and it is $81.88\,\mathrm{s}$ with an

Navigate corridor	Task: Technical, Human: Reliable
Enter patient's room	Task: Technical, Human: Average
Monitor patient	Task: Technical, Human: Fragile
Leave room	Task: Interaction, Human: Fragile

Table 8: Parameters for an emergency

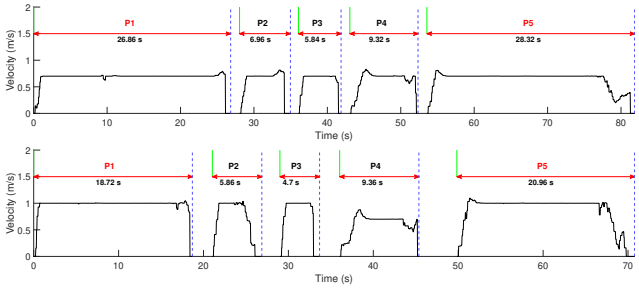

Figure 6: Velocity profiles and times for navigation phases (red: corridor ones) in the emergency scenario. Top: *cohan*, Bottom: *cohan+platinum*

overhead time of $0.92\,\mathrm{s}$ in the *cohan* setting. This clearly shows that the integrated approach takes lesser time to reach the patient in an emergency even with a larger overhead. Moreover, the execution time for the *cohan* setting is $77.30\,\mathrm{s}$ which is larger than the overall planning time of the integrated approach, whereas, in the *cohan+platinum* setting, it is only $59.60\,\mathrm{s}$. The benefit of the proposed TAMP approach is apparent from this scenario.

Drug Delivery Scenario The scenario simulates a robot that has to take the prescribed drugs from the pharmacy and deliver them to a patient in a room. The complete set of parameters for this scenario is listed below in Table. 9.

Navigate corridor	Task: Technical, Human: Average
Enter pharmacy	Task: Interaction, Human: Average
Approach pharmacist	Task: Interaction, Human: Reliable
Leave pharmacy	Task: Interaction, Human: Average
Enter patient's room	Task: Social, Human: Average
Deliver drugs	Task: Interaction, Human: Fragile
Leave room	Task: Social, Human: Average

Table 9: Parameters for drug delivery

Fig. 7 shows the velocity profiles and the times for each phase. The corridor navigation phases in this scenario are P1, P5, and P9, where it can be seen that the robot with dynamic parameter adaptation takes lesser time to traverse. Phases P2 to P4 are the interaction phases of the robot with the pharmacist who might have already interacted with the robot and could act reliably compared to other humans in the environment. Therefore, the robot could use nominal human-aware parameters during these phases. Compared to the patrolling scenario, instead of monitoring the human, the robot has to interact with the patient in P7. Hence, the robot adapts a nominal velocity to be more responsive while interacting in the *cohan+platinum* setting.

The *cohan* setting takes $114.46\,\mathrm{s}$ to complete the task while the *cohan+platinum* setting takes $124.46\,\mathrm{s}$. However,

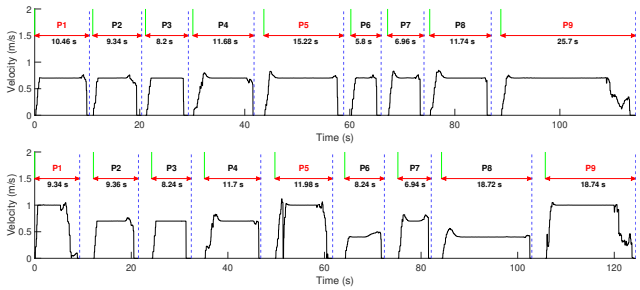

Figure 7: Velocity profiles and times for navigation phases (red: corridor ones) in the drug delivery scenario. Top: *cohan*, Bottom: *cohan+platinum*.

the execution time for the *cohan+platinum* setting is lesser ($103.26\,\mathrm{s}$) than the *cohan* setting ($105.1\,\mathrm{s}$). Hence, the robot effectively spends lesser time in navigation and delivery while using *cohan+platinum* configuration. Although this initial integration has an overhead time of $2.35\,\mathrm{s}$ per phase in the *cohan+platinum* setting, it could be reduced systematically.

Final Remarks and Future Works

We presented a novel integrated Task and Motion Planning approach to enhance the awareness of the social navigation skills of robots. We discussed task-related and human-related knowledge leveraged by a task planner to dynamically reconfigure the parameters of a motion planner and adapt the planning of trajectories to different (social) situations. We evaluated and tested the proposed approach in a simulated assistive domain, designing three different scenarios that required the robot to interact with different types of humans and perform various types of tasks. Note that this approach assumes perfect tracking of humans while planning and the types of humans are known beforehand to the task planner. When compared to a situation in which the robot uses the motion planner alone, results show that the integrated approach effectively allows the robot to dynamically adapt the velocity and motion behaviors to perform better in varying social situations.

Future works will focus on improving the efficiency of the approach by reducing the overhead detected during the experiments. On a longer time horizon, we plan to investigate the integration also of augmented perception capabilities to enrich the contextual knowledge provided online by the task planner. Namely, perception capabilities would enrich dispatched motions by providing information about detected social situations (e.g., detecting a fragile human instead of a reliable human expected at planning time).

Acknowledge

LAAS-CNRS authors are partially supported by the European Union's Horizon Europe euROBIN project under grant agreement No 101070596. CNR authors are partially supported by Italian MUR and the PNRR research project "Fit4MedRob- Fit for Medical Robotics" - "Piano Nazionale Complementare D.D. n. 931 6/6/2022 Cod. PNC0000007"

References

Allen, J. F. 1983. Maintaining knowledge about temporal intervals. *Commun. ACM* 26(11):832–843.

Awaad, I.; Kraetzschmar, G. K.; and Hertzberg, J. 2015. The Role of Functional Affordances in Socializing Robots. *International Journal of Social Robotics* 7(4):421–438.

Beraldo, G.; Koide, K.; Cesta, A.; Hoshino, S.; Miura, J.; Salvà, M.; and Menegatti, E. 2022. Shared autonomy for telepresence robots based on people-aware navigation. In Ang Jr, M. H.; Asama, H.; Lin, W.; and Foong, S., eds., *Intelligent Autonomous Systems 16*, 109–122. Cham: Springer International Publishing.

Bruno, B.; Recchiuto, C. T.; Papadopoulos, I.; Saffiotti, A.; Koulouglioti, C.; Menicatti, R.; Mastrogiovanni, F.; Zaccaria, R.; and Sgorbissa, A. 2019. Knowledge representation for culturally competent personal robots: Requirements, design principles, implementation, and assessment. *International Journal of Social Robotics* 11(3):515–538.

Cavallo, F.; Semeraro, F.; Fiorini, L.; Magyar, G.; Sinčák, P.; and Dario, P. 2018. Emotion modelling for social robotics applications: A review. *Journal of Bionic Engineering* 15(2):185–203.

Che, Y.; Okamura, A. M.; and Sadigh, D. 2020. Efficient and trustworthy social navigation via explicit and implicit robot–human communication. *IEEE Transactions on Robotics* 36(3):692–707.

Cialdea Mayer, M.; Orlandini, A.; and Umbrico, A. 2016. Planning and execution with flexible timelines: a formal account. *Acta Informatica* 53(6-8):649–680.

Clodic, A., and Alami, R. 2021. *What Is It to Implement a Human-Robot Joint Action?* Cham: Springer International Publishing. 229–238.

Clodic, A.; Pacherie, E.; Alami, R.; and Chatila, R. 2017. *Key Elements for Human-Robot Joint Action*. Cham: Springer International Publishing. 159–177.

Cortellessa, G.; Benedictis, R. D.; Fracasso, F.; Orlandini, A.; Umbrico, A.; and Cesta, A. 2021. AI and robotics to help older adults: Revisiting projects in search of lessons learned. *Paladyn, Journal of Behavioral Robotics* 12(1):356–378.

Ferrer, G.; Zulueta, A. G.; Cotarelo, F. H.; and Sanfeliu, A. 2017. Robot social-aware navigation framework to accompany people walking side-by-side. *Autonomous Robots* 41(4):775–793.

García-Betances, R. I.; Cabrera-Umpiérrez, M. F.; Ottaviano, M.; Pastorino, M.; and Arredondo, M. T. 2016. Parametric cognitive modeling of information and computer technology usage by people with aging- and disability-derived functional impairments. *Sensors* 16(2).

Ingrand, F., and Ghallab, M. 2017. Deliberation for autonomous robots: A survey. *Artificial Intelligence* 247:10–44.

Kostavelis, I.; Vasileiadis, M.; Skartados, E.; Kargakos, A.; Giakoumis, D.; Bouganis, C.-S.; and Tzovaras, D. 2019. Understanding of human behavior with a robotic agent through daily activity analysis. *International Journal of Social Robotics* 11(3):437–462.

Kruse, T.; Pandey, A. K.; Alami, R.; and Kirsch, A. 2013. Human-aware robot navigation: A survey. *Robotics and Autonomous Systems* 61(12):1726–1743.

Lemaignan, S.; Warnier, M.; Sisbot, E. A.; Clodic, A.; and Alami, R. 2017. Artificial cognition for social human–robot interaction: An implementation. *Artificial Intelligence* 247:45–69. Special Issue on AI and Robotics.

Mead, R., and Matarić, M. J. 2017. Autonomous human–robot proxemics: socially aware navigation based on interaction potential. *Autonomous Robots* 41(5):1189–1201.

Moro, C.; Nejat, G.; and Mihailidis, A. 2018. Learning and personalizing socially assistive robot behaviors to aid with activities of daily living. *ACM Trans. Hum.-Robot Interact.* 7(2):15:1–15:25.

Repiso, E.; Garrell, A.; and Sanfeliu, A. 2022. Adaptive social planner to accompany people in real-life dynamic environments. *International Journal of Social Robotics*.

Rios-Martinez, J.; Spalanzani, A.; and Laugier, C. 2015. From proxemics theory to socially-aware navigation: A survey. *International Journal of Social Robotics* 7(2):137–153.

Rossi, S.; Ferland, F.; and Tapus, A. 2017. User profiling and behavioral adaptation for HRI: A survey. *Pattern Recognition Letters* 99:3 – 12.

Singamaneni, P. T.; Favier, A.; and Alami, R. 2021. Human-aware navigation planner for diverse human-robot interaction contexts. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5817–5824.

Singamaneni, P. T.; Favier, A.; and Alami, R. 2022. Watch out! there may be a human. addressing invisible humans in social navigation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 11344–11351. IEEE.

Triebel, R.; Arras, K.; Alami, R.; Beyer, L.; Breuers, S.; Chatila, R.; Chetouani, M.; Cremers, D.; Evers, V.; Fiore, M.; Hung, H.; Ramírez, O. A. I.; Joosse, M.; Khambhaita, H.; Kucner, T.; Leibe, B.; Lilienthal, A. J.; Linder, T.; Lohse, M.; Magnusson, M.; Okal, B.; Palmieri, L.; Rafi, U.; van Rooij, M.; and Zhang, L. 2016. *SPENCER: A Socially Aware Service Robot for Passenger Guidance and Help in Busy Airports*. Cham: Springer International Publishing. 607–622.

Umbrico, A.; Cesta, A.; Cialdea Mayer, M.; and Orlandini, A. 2017. PLATINUm: A New Framework for Planning and Acting. *Lecture Notes in Computer Science* 498–512.

Umbrico, A.; Cesta, A.; Cialdea Mayer, M.; and Orlandini, A. 2018. Integrating resource management and timeline-based planning. In *The 28th International Conference on Automated Planning and Scheduling (ICAPS)*.

Umbrico, A.; Cesta, A.; Cortellessa, G.; and Orlandini, A. 2020. A holistic approach to behavior adaptation for socially assistive robots. *International Journal of Social Robotics* 12(3):617–637.