A Humanoid Robot Asks Humans for Help to Navigate Elevators

Niklas Arlt, Fabian Reister, Christian Dreher and Tamim Asfour

***Abstract*— Autonomous elevator navigation represents a critical challenge for service robots operating in multi-floor environments. This paper presents a novel framework that integrates autonomous elevator operation with human assistance, enabling robots to navigate elevators in diverse scenarios with varying human presence. Our approach determines whether autonomous operation is feasible based on real-time environmental constraints and reactively switches to seeking human help when necessary. We demonstrate how combining different navigation cost metrics allows the robot to navigate safely among humans and reliably detect door states based on LiDAR data, even with humans entering or exiting the elevator. We validate our system through comprehensive testing in both simulated human-robot interaction scenarios and real robot experiments using the humanoid household robot ARMAR-7. Results show significantly improved success rates across diverse elevator situations compared to pure autonomous or help-seeking baselines.**

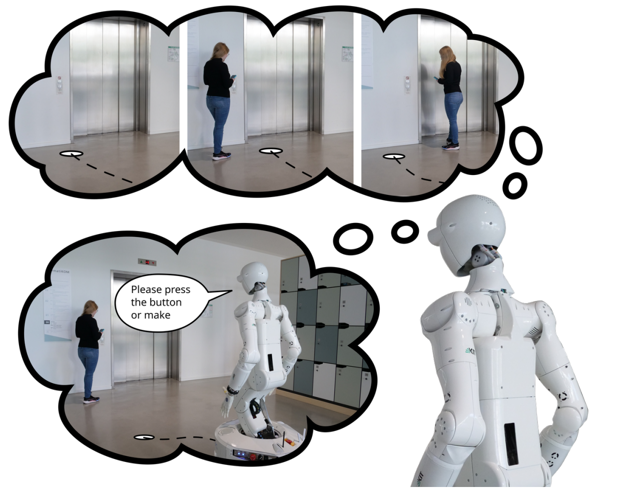

Fig. 1. The humanoid robot ARMAR-7 dynamically selects between autonomous elevator operation and requesting help. It considers humans for finding suitable poses for button pressing (top) and approaches humans and talks to them to convey its help intent (bottom).

I. INTRODUCTION

As robots continue to be integrated into human-centered environments, their capacity to autonomously navigate multi-floor buildings, e. g., for fetch-and-delivery tasks, becomes increasingly essential. Navigating between floors via elevators represents a critical capability for service robots, yet presents significant challenges in environments without specialized robot interfaces. Unlike navigation on a single floor, elevator operation demands complex sequences of actions – summoning the elevator, entering, selecting the destination floor, and exiting – while adhering to social compliance and collision avoidance in confined spaces designed primarily for human use. These challenges are amplified in public environments with dynamic human presence, where robots must adapt their strategies based on the current state of the environment.

Veloso et al. ([1], [2]) introduced the paradigm of *symbiotic autonomy*, wherein robots understand their limitations and proactively seek human assistance to overcome them. Their CoBots requested help for tasks beyond their capabilities, such as pressing elevator buttons or opening doors. This approach, while effective when humans are available, leaves robots dependent on assistance and potentially stranded when humans are absent. We propose a hybrid approach that combines autonomous operation with human assistance. Our system continuously evaluates environmental conditions to determine the optimal strategy: operating the elevator independently when possible, or engaging with humans when autonomous operation is hindered. Compared to Veloso et al. [2], our approach offers advantages. While their system relies exclusively on human assistance, our robot can successfully navigate elevators both when humans are available to help and when it is alone. This eliminates the substantial waiting periods documented in [2], where robots would remain idle for extended durations until human assistance became available. By integrating autonomous capabilities with human interaction, our system maintains continuous operation across varying human presence scenarios, addressing a critical limitation in previous elevator navigation systems. Even in case the robot is able to operate the elevator autonomously, the robot needs to be aware of humans to avoid collisions while navigating or pressing the buttons. Thus, the robot must constantly monitor and adapt to human movements to prevent collisions in these confined spaces.

Beyond human awareness, the *affordance spaces* – defined as "social spaces related to potential activities provided by the environment" [3] – play a central role in effective elevator operation. These spaces directly correspond to the specific activities required for elevator use, such as accessing button panels or positioning inside the elevator for transport between floors. Human presence significantly constrains these available affordance spaces, creating a dynamic decision framework for autonomous operation. The robot can leverage this spatial awareness to identify appropriate positions

This work has been supported by the Carl Zeiss Foundation through the JuBot project, by the Baden-Württemberg Ministry of Science, Research and the Arts (MWK) as part of the state's "digital@bw" digitization strategy in the context of the Real-World Lab "Robotics AI", and the German Federal Ministry of Research, Technology and Space (BMFTR) under the Robotics Institute Germany (RIG).
The authors are with the High Performance Humanoid Technologies Lab, Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology (KIT), Germany. E-mails: {niklas.arlt, asfour}@kit.edu

for button interaction, or importantly, to determine when autonomous operation is infeasible due to spatial constraints. This affordance-based reasoning enables more socially aware decision-making about when to proceed autonomously versus when to request human assistance. To address these challenges, we extend the work in [4], which calculates optimal robot placements for mobile manipulation tasks, to affordance spaces relevant for elevator use.

Given that the viability of autonomous operation fluctuates with dynamic human positioning, we propose an adaptive approach that integrates behavior trees into the robot's cognitive architecture. This integration enables reactive switching between autonomous operation and help-requesting strategies based on real-time interpretation of the scene. The behavior tree framework systematically manages the execution of autonomous skills, continuously monitors their execution progress, and dynamically adapts subsequent goals based on current environmental conditions. This reactive behavior ensures the robot can seamlessly transition between operational modes when affordance spaces become constrained or available, optimizing both task completion and social appropriateness in shared elevator environments.

We evaluate the effectiveness of our approach using a set of scenarios to assess the robot's performance in elevator-taking in the presence of humans. The scenarios consider multiple human behaviors, initial positioning and human goals. We compare our approach to baselines using this framework and evaluate its performance in real robot experiments on a humanoid household robot.

II. Related Work

A. Robot Skills for Autonomous Elevator Use

Navigating elevators with robots has been of interest to the community for a long time. While most of the work focused on robots without the ability to press the buttons ([1], [2], [5], [6]), humanoid robots ([7], [8]) and robots specifically built for navigating elevators have also been used ([9], [10], [11]). While many aspects like button detection ([12], [13]), elevator button pressing ([7], [11]), elevator door state estimation ([5], [9]), floor number recognition [8], navigation ([14], [15]), as well as entering and exiting the elevator in the presence of humans ([16], [17]) have been studied, no work so far has focused on the options for a robot that can navigate the elevator autonomously but still ask humans for help.

For the button detection, recent works ([18], [12], [13]) use neural networks trained (additionally) on an elevator button dataset to detect buttons. One option is to then use an optical character recognition approach to detect the button label [12]. By using a depth camera, these detected buttons can be converted into 3D point clouds ([19], [7]). Button pressing is usually done by moving the robotic arm to the button pose or slightly behind it ([19], [9], [7]). Compared to our robot, which features humanoid hands, past work used either grippers [7] or even simpler end-effectors [9].

The elevator door state can be detected reliably with LiDAR sensors ([9], [5], [20]), e. g.,, by calculating an opening angle [20] or determining if the door space is occupied by obstacles [5]. However, existing methods fail to differentiate between door occlusion by humans and actual door closure, leading to false closed-door detection when humans occupy or traverse the doorway.

Regarding the necessary navigation, past work has mostly focused on entering the elevator in the presence of humans ([17], [16], [6]). After humans are detected, either an occupancy grid is constructed ([6], [17]) or a learned controller is directly used for navigating into the elevator respecting the humans [16]. With the occupancy grid, different navigation approaches have been employed, namely potential field navigation [17] and a human-aware navigation approach ([6], [15]).

B. Interaction With Humans in Elevator Use

One focus of past work on taking elevators has been interactions with humans. Humans were asked to press elevator buttons and hold the door open for the robot ([1], [2]). Other work has addressed navigation into the elevator ([16], [17]) or initiating communication when humans are blocking the doorway [21]. Recent work also incorporated social-aware navigation ([6], [20]) to ensure proper distance to humans and to respect affordance spaces ([22], [23]). The used algorithms were quite simple, especially compared to the numerous human-aware capabilities found in [24]. One way of representing distances to humans is the Proxemics model developed by Hall [25] with an intimate, personal, social, and public zone. However, even keeping a personal distance of $0.5\,\mathrm{m} - 1\,\mathrm{m}$ can block the robot from entering most occupied elevators due to the limited space, even though it would still fit inside. For robots operating close to humans, signaling the robot's intent to human bystanders is important [26]. In [27], a robot used eye gaze to guide humans in identifying locations on a map. This suggests that eye gaze can be used to convey whom the robot is addressing or which button the robot intends to press. In [28], vocal cues were found to improve the impression of the robot while entering and exiting the elevator. In unclear situations, like when humans are trying to exit the elevator while the robot wants to enter, voice cues could similarly be used to convey the robot's intent to let the humans exit first.

C. Coordination and Task Planning

State-of-the-art elevator operation methods typically involve three steps: the robot navigates to the control panel or entrance, requests the elevator via button press or direct communication, and keeps the door open while entering ([7], [19], [10], [17], [5], [21]). Some approaches include additional tasks like determining the elevator travel direction [17] or confirming the robot is on the correct floor ([19], [20]). Current works that consider human collaboration to improve the robot's reliability either depend on direct intervention ([1], [2]) or require elevator modification to control the elevator via an API [21]. We address the shortcomings of both approaches by integrating human assistance into a fully autonomous robot while maintaining the reliability benefits.

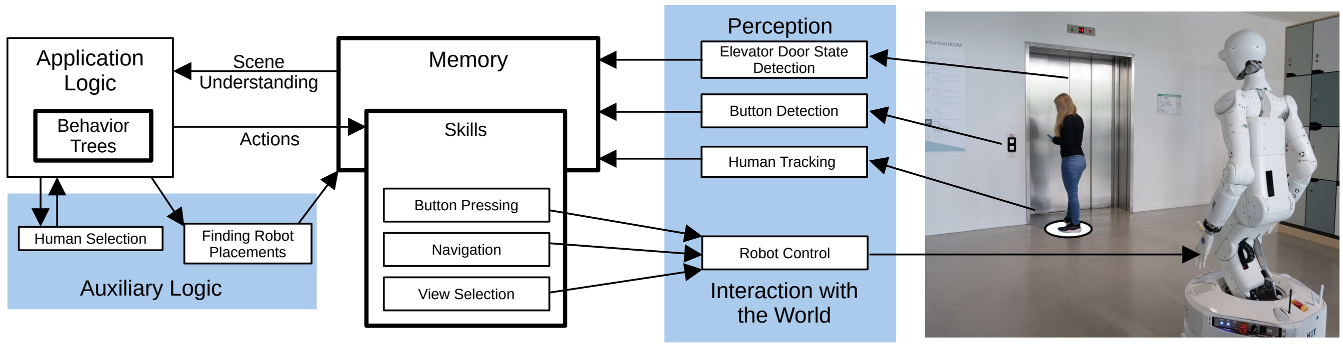

Fig. 2. System overview including the three main components behavior trees, memory and skills which are supported by auxiliary logic for human selection and platform placement calculation. The memory is the single source of truth which gets the data from perception modules for elevator door state, buttons and human tracking. Memory-based skills like button pressing, navigation and view selection are used to control the robot.

Compared to [2], our robot has all the skills required to navigate the elevator autonomously. The authors in [2] focus on human-assisted operation, where the robot always requests help and proactively seeks humans to minimize the waiting time. Their main contribution includes this help-seeking strategy and a user study on who to ask for help. We instead only rely on human communication when autonomous operation fails due to unforeseen circumstances, such as humans blocking access to buttons, blocking the robot's path through the doorway when entering, exiting the elevator when the robot wants to enter, or preventing the robot from exiting. Therefore, our approach does not need to identify individual people, and the area of interest to our approach only covers the direct proximity to the elevator, as opposed to everything accessible to the robot on the current floor.

A critical requirement is rapid reactivity to dynamic human behavior. While recent works employ finite state machine variants ([9], [6], [20], [7]), behavior trees offer superior reactivity and better scalability [29], making them more suitable for dynamic environment requirements.

III. Approach

The system architecture is shown in Fig. 2. It builds upon the memory-centered cognitive architecture in ArmarX [30] and allows for easy integration of our elevator- and human-interaction-specific modules. The high-level behavior coordination using behavior trees enables rapid responses to dynamic changes in the world state. It leverages memory information to make decisions and coordinates the robot's actions through a memory-based skill framework. As shown in Fig. 2, elevator-specific perception data – including human tracking, door state detection, and button detection with pose estimation – is maintained in memory. The system primarily utilizes skills for button pressing, navigation, and gaze control, complemented by auxiliary logic for robot placements and human selection for interaction.

A. Behavior Coordination Using Behavior Trees

Behavior trees are used to (i) reason about the current state to reactively adapt the behavior and (ii) monitor the autonomous execution to identify the necessity to ask for help. Depending on the perceived environment and robot state queried from the memory, the behavior tree's execution switches between different autonomous help-requesting strategies to achieve the next task.

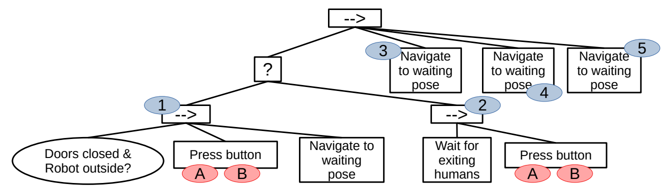

Fig. 3. Top-level behavior tree. All leaf nodes are implemented as sub-trees. → indicates sequences, ? indicates selections. 1: Sub-tree to call elevator, 2: Sub-tree to navigate inside and press button, A: Includes autonomous strategy with navigation to button, B: Includes help-seeking strategy with approaching of human. Decisions are made on perception (door opened/closed), robot position, success of navigation-planner (humans in the way), whether or not the button is detected, etc.

One challenge is the abstraction of all necessary perception, processing, and controlling tasks into behaviors that can be used in a behavior tree. We chose an approach where each behavior that controls the robot keeps track of its own progress in each tick and stays in a running state until it finishes. Since most perception and processing tasks require minimal runtime, they immediately return results that can be used directly in behavior tree decisions.

Unlike previous work exploring diverse elevator interaction approaches, we focus specifically on unmodified elevators controlled solely through button interfaces.

To enable robots to take the elevator autonomously, we find the following relevant tasks: (i) Ensure the outside button is pressed, (ii) wait for the elevator doors to open, (iii) move into the elevator – including waiting for humans to exit the elevator, (iv) ensure the button for the destination floor is pressed and (v) move out of the elevator.

Fig. 3 shows the high-level logic that the robot follows. Compared to the sequential execution of the before-mentioned tasks, we enhance the robot's flexibility via a selector node between sub-trees 1 and 2 to switch between tasks (i) - (iv). This allows the robot to skip pressing the outside button when doors are already open and retry if doors close unexpectedly during entry.

Similar to [1], our robot only asks humans for help in

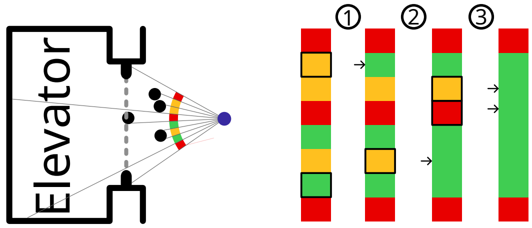

Fig. 4. LiDAR-based door state detection with the robot (blue circle) and humans (black circles) in front of the door and inside the doorway. The three processing steps allow for an accurate door state detection, provided measurements near the door center remain unoccluded. ①: Ensure symmetry since both doors are coupled, ②: Replace unknown cells surrounded by free cells, ③: Remove occupied cells that are separated from the sides by a free cell. Green cells indicate free (obstacles only behind the doorway), orange cells indicate unknown (obstacle in front of the doorway), and red cells indicate occupied (obstacles detected in the doorway).

case it cannot achieve its goal autonomously. However, the additional ability to control the elevator by pressing buttons allows our robot to require less human interaction. The only cases where additional interaction abilities are needed are when (i) humans block the affordance space for button pressing, (ii) humans block the doorway by a) standing inside the elevator but too close to the doorway, b) entering or c) exiting the elevator, (iii) the robot (temporarily) cannot press the button or (iv) the robot needs to let people exit the elevator first to avoid mutual blocking.

Most tasks in Fig. 3 can be achieved by an autonomous and a help-requesting strategy. The autonomous strategy is preferred in all cases to minimize unnecessary human interaction, only switching to the help-requesting strategy in cases of failure. For navigation, this means that the robot first attempts to navigate to its goal autonomously. If this fails because humans block all possible paths to the door, the robot approaches them to signal its intent to pass, and if they still don't move, verbally requests them to clear the pathway. For button pressing, the robot first attempts to navigate to the button autonomously. When humans stand too close to the button, it requests the nearest human to the button to either move aside or press the button.

B. Key Modules for Elevator Use

Key modules for autonomous execution include button affordance extraction, button push action generation, and base placements determination for those actions.

Perception: The robot's perception module needs to detect (i) buttons to extract push affordances and press the buttons, (ii) whether the elevator door is open or closed, and (iii) humans to avoid collisions and let them exit the elevator. The robot uses an articulated elevator model with approximate button positions to position itself and visually localize the buttons. We use OCR-RCNN [18] to extract button bounding boxes from the RGB image and extract the button label from the cropped image. The button pose is then estimated by calculating the normal vector to the button plane after estimating the button plane from the cropped point cloud, similar to [7].

Unlike other door state detection approaches, our method accounts for humans walking through the doorway. We extend the approach from [5] by deploying additional virtual boxes inside the doorway, introducing a third occupation state "unknown" for occluded boxes to the already existing "occupied" and "free" states, and applying post-processing to the occupation data. Since our elevator doors close symmetrically from both sides, we assume that both doors open to the same degree simultaneously. This allows us to infer the state of occluded boxes on one side by checking the corresponding boxes on the other side. Additionally, occupied and occluded boxes that are not connected to the side cannot belong to the door and are therefore treated as free boxes in the opening degree calculation, enabling accurate detection even when humans occupy the doorway. Fig. 4 illustrates the processing.

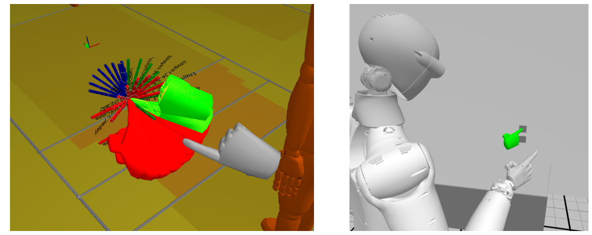

Fig. 5. Hypothesis generation for push affordance using finger TCP (left) and execution of best hypothesis (right). Red hand poses are rejected (e. g., no IK-solution), good hand poses in green.

After estimating human poses, we track them by clustering the LiDAR points and associating clusters with the previous human position estimate if the size matches and the distance is small enough. The human's velocity vector is then calculated from its temporally smoothed position. Exiting humans are detected when their velocity vector direction opposes the robot's movement direction during elevator entry attempt.

Button pressing: Given a detected button, we generate multiple action candidates and select the best one. Each candidate defines a press pose consisting of hand shape and the index finger tip position (Tool Center Point, TCP) located slightly behind the button center. As depicted in Fig. 5, each pose is characterized by (i) the rotation angle around the button surface normal and (ii) the hand pitch angle.

The pitch angle prevents inward finger bending that would break contact, while the rotation angle is optimized based on the robot's base position relative to the button and button height.

Determine affordance spaces: The goal is to identify optimal base poses for action execution. Target actions include (i) perceiving and pressing elevator buttons (both the outside and inside), (ii) positioning within or in front of the elevator, and (iii) approaching humans to ask for help. Determining action feasibility, i. e., whether required spaces are available or blocked by humans, is crucial for high-level behavior planning. To address these requirements, we employ a costmap-based approach, which extends our previous work described in [4]. The objective is to find the robot placement that minimizes the total cost c:

$$c = c_n + c_m + c_p + c_s \quad (1)$$

Similar to [4], the total cost comprises (i) navigation costs c_n representing travel distance and obstacle proximity,

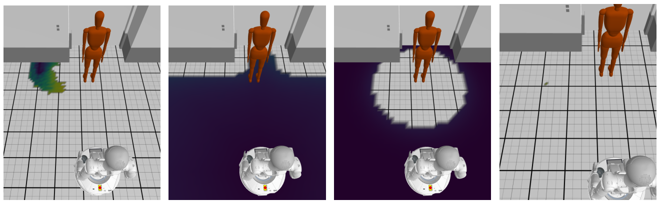

Fig. 6. Left to right: Manipulation costs, navigation costs, social costs and combined costs. The combined costs only leave a very small patch for the robot to the left of the human. Colorscheme: Viridis (increasing costs from violet to blue, green, yellow).

(ii) manipulation costs c_m based on inverse reachability maps to ensure kinematic feasibility and high end-effector manipulability. Additionally, we introduce (iii) a placement cost c_p that restricts placements to ensure adequate button visibility, and (iv) a social cost c_s that enforces socially compliant placement (see Fig. 6).

As the robot operates in close proximity to humans, its motion should be human-like. Therefore, we modify the generation of the reachability maps by only sampling the robot's joint configurations that lie within the human joint limits [31], [32]. Central to social compliance are proxemics for humans $\mathcal{H}$. Following, we employ a symmetric Gaussian function when human velocity estimation is not available:

$$c_{\text{prox}}(\boldsymbol{p}) = \max_{h \in \mathcal{H}} \left(\exp \left(- \frac{||\boldsymbol{p_h} - \boldsymbol{p}||^2}{2\sigma^2_{\text{prox}}} \right) \right). \tag{2}$$

Within the personal space ($||\boldsymbol{p_h} - \boldsymbol{p}|| < d_{\text{pers}}$), we invalidate the costmap to ensure human comfort [23]. The social cost term c_s is task-dependent and is determined as follows: (i) Button pressing: $c_s = c_{\text{prox}}$ with manipulation costs derived from inverse reachability maps for both execution and pre-pose. (ii) Elevator positioning: $c_s = \zeta_{\text{center}} + \zeta_{\text{clear button}} + \zeta_{\text{prox}}$, where $\zeta_i = w_i c_i$ are weighted costs and c_{center} favors central positioning inside the elevator while $c_{\text{clear button}}$ prevents the robot from blocking button access. (iii) Human approaching: $c_s = \zeta_{\text{prox}} + \zeta_{\text{interact}}$, where c_{interact} maintains an appropriate interaction distance to the human and favors alignment along the button-human axis.

Human-aware navigation: For each navigation target, we plan a global path and employ the Timed Elastic Bands [15] local planner to react to dynamic changes such as moving humans. Preliminary tests showed that recalculating the robot placement at every behavior tree cycle to respond to moving humans caused oscillations during human approach due to small shifts in their detected position. Therefore, we recalculate the robot placement only when the current navigation goal becomes occupied.

IV. Evaluation

We evaluate our approach against several baselines in simulation and real-world tasks and assess its performance using the humanoid robot ARMAR-7.

A. Baselines

1) *Naïve baseline – No asking for help*: Human interaction is not incorporated into the behavior tree level. Humans are only considered to find robot base placements and not for, e. g.,, asking them to make space or letting them exit the elevator first. The robot only tries the autonomous strategies for each task.
2) *Veloso et al.– No manipulation*: Similar to the work of Veloso et al. [1], the robot does not operate the elevator by pressing buttons and instead relies on humans for doing that. This results in the robot having different navigation goals as it can, e. g., stand anywhere in the elevator and ask humans to press the button, instead of having to stand directly in front of the button.

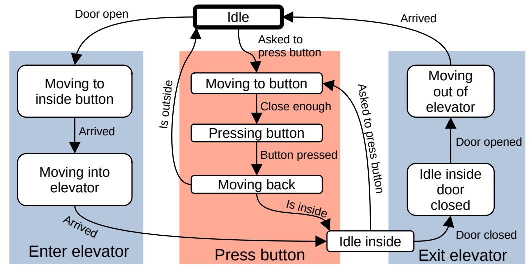

Fig. 7. State machine describing the behavior of a human who enters the elevator (left), exits the elevator (right), and presses buttons if asked by the robot (middle) used in our simulation environment. The initial state (Idle) is marked with a thicker outline.

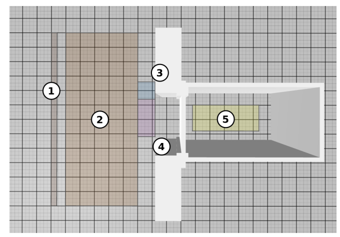

Fig. 8. Sampling areas for simulated humans shown on the elevator model (white) from above. ① Goal for exiting humans, ② standing humans in front of the elevator, ③ in front of the button panel, ④ waiting directly in front of the elevator, and ⑤ inside the elevator.

B. Evaluation Environments

Simulation environment: To compare our approach against the baselines in different elevator situations, we use a kinematic simulation. The elevator door is simulated to open after one of the buttons is pressed and to close delayed after the last time someone was in the doorway. The door only closes after the robot goes through the door, allowing us to measure the time it needs to stay open based on the robot's entering speed. For the simulated human movement, we use "optimal reciprocal collision avoidance" [33], and for the general behavior of the humans, we employ a state-based structure for each human's goal. These goals include entering the elevator, exiting the elevator, staying at the current location, and both entering and exiting. Fig. 7 shows the modeled behavior for the latter.

To randomize each experiment, we sample the human positions for each human and state. The sampling is done within certain areas with a meaning associated with them (e. g., inside the elevator, in front of the elevator, directly in front of the door, and in front of the outside button) as shown

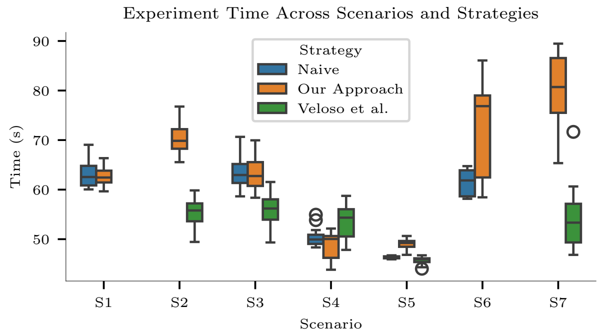

Fig. 9. Execution time in simulation for each approach on each scenario. The Veloso et al. baseline is much faster in some scenarios because we expect humans to be faster in moving to the button and pressing it.

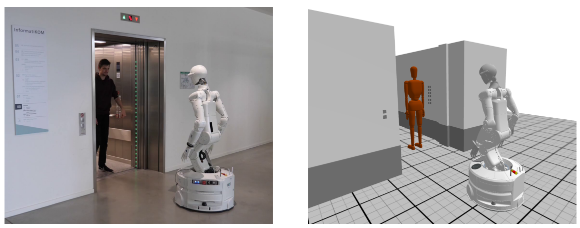

Fig. 10. The real elevator (left) and the simulation counterpart (right).

in Fig. 8. Different states in the human's state machines have different locations associated with them, which are all sampled from one of the areas. We supply a generator for experiment descriptions with sampled human positions through our paper webpage[1]. When the robot asks humans to make space or press a button, we assume cooperative behavior, whenever possible (humans cannot move through closed doors). For all scenarios, the robot's initial pose is sampled from a line similar to area ①. Each experiment was conducted 20 times.

Real robot environment: We evaluate our approach with the humanoid household robot ARMAR-7. The evaluation environment includes the space in front of an elevator and the elevator itself (see Fig. 10). The elevator is around $3\,\mathrm{m}$ long and $1.6\,\mathrm{m}$ wide with a door width of $1.3\,\mathrm{m}$. A model of the elevator is available to the robot, including the approximate locations of elevator buttons and annotated regions (inside the elevator, doorway, area outside the elevator where the robot can wait).

C. Evaluation Scenarios

Scenario S1 – No humans: This scenario shows how each approach works without any human help.
Experimental Setup: There are no humans in the scene, and the door is initially closed without any pressed buttons.
Results – Simulation: In simulation, both the *naïve baseline* and our approach achieve near-perfect success. Both approaches result in the same behavior in this scenario and need around $65\,\mathrm{s}$ as shown in Fig. 9. As expected, the *Veloso et al. baseline* never succeeds in this scenario.
Results – Real Robot: While we needed to hold the doors open for the robot, it was still able to do everything else, including detecting and pressing the buttons and navigating to all necessary places (see video for details).

Scenario S2 – Human blocking the outside buttons: This scenario shows how each approach handles blocked access to the buttons.
Experimental Setup: There is one human standing in area ③ blocking access to the outside button panel. The door is initially closed, and no buttons are pressed.
Results – Simulation: Our experiments show that the *naïve baseline* does not work in this scenario, while the other two approaches have a high success rate over $90\,\%$. The *Veloso et al. baseline* is about $15\,\mathrm{s}$ faster than our approach, because it asks the human to press the button, which it can do without even needing to first move towards the button, as it already is standing there. Our approach requests the human to move aside so the robot can press the button, which requires additional time for both human repositioning and button pressing, resulting in a total time of about $70\,\mathrm{s}$.
Results – Real Robot: We tested this scenario in two configurations: a human standing directly in front of the button and standing approximately $1\,\mathrm{m}$ to the side. When the human blocked direct button access, the robot successfully approached and requested space in most trials. After the human complied, the robot acknowledged with "thank you," approached the button, and resumed pressing. When the human stood to the side, the robot navigated to the opposite side, enabling safe button operation without human interaction.

Scenario S3 – Human providing help in front of elevator: The goal of this scenario is to see how the robot behaves when humans are close to the elevator, but not directly blocking access to the doorway or buttons.
Experimental Setup: One human is standing in area ② without intending to take the elevator. Initially, the door is closed and no buttons are pressed.
Results – Simulation: All approaches achieve high success rates. The *Veloso et al. baseline* is about $5\,\mathrm{s}$ faster, because we assume that humans are much faster than the robot in pressing the buttons.
Results – Real Robot: For our approach, the robot's actions are identical to S1.

Scenario S4 – Human waiting for elevator: The goal of this scenario is to demonstrate how the approaches respond when the doors open without pressing the buttons and to measure the time savings from bypassing button pressing.
Experimental Setup: One person is standing in area ④ in front of the elevator door. The door is closed, but the button is already pressed.
Results – Simulation: All approaches work well in this case. The *Veloso et al. baseline* is a bit slower at around $55\,\mathrm{s}$ compared to $50\,\mathrm{s}$ for the other approaches.
Results – Real Robot: Since the robot's vision currently cannot determine if the button is already pressed, the robot always moved to the left side of the button and occasionally initiated pressing. While navigating to the button or pressing it, the robot noticed that the door opened and started navigating into the elevator, interrupting its previous actions.

[1] https://sw.pages.h2t.iar.kit.edu/robot-asks-humans-for-help/site/

After that, it positioned itself at an appropriate location inside the elevator while avoiding collisions with the human or the elevator. When the human was standing in the back, the robot moved to the inside button panel and pressed the button. When the human was standing at the button panel, the robot instead moved to the back of the elevator and asked the human to press the button. This occasionally caused oscillatory robot movements between positions, resulting in human tracking failures.

Scenario S5 – Human leaving elevator: This scenario shows how the approaches react to humans leaving the elevator and how important it is to wait for them.

Experimental Setup: One human is standing in area ⑤ and moves to area ① as soon as the elevator door opens. The door is closed, but the button is already pressed.

Results – Simulation: All approaches showed a high success rate except for the *naïve baseline*. As expected, all three approaches were fast (less than 50 s). Unexpectedly, all approaches also took almost the same time. This means that – in this scenario – the time for pressing the inside button and asking the human to press the button was similar.

Results – Real Robot: We expected that the robot asks for humans to exit the elevator and positions itself in front of the door. Here, the robot stopped after the humans tried to exit, but it sometimes missed the voice cue and was a bit late in going back in front of the elevator for waiting.

Scenario S6 – Human as co-rider: This scenario shows how well each approach works if other humans are already inside the elevator – especially regarding the entering.

Experimental Setup: One human is standing in area ⑤ and stays in this area for the entire experiment. Initially, the door is closed and the button is not pressed.

Results – Simulation: Our simulation only shows very good results for our approach and a success rate around 30% for the *naïve baseline*. This is because the robot is not able to get into the elevator when the human is standing too close to the doorway, if it does not ask the human to make more space so that the robot can enter.

Results – Real Robot: Similar results to S4.

Scenario S7 – Combined: The goal is to show how the approaches handle more crowded situations.

Experimental Setup: One human is in area ⑤ and moves to area ① when the doors open. One human is in area ④ and moves to area ⑤. Another human in area ③ is blocking access to the buttons. The door is initially closed, and the button is not pressed.

Results – Simulation: The results look very similar to S2, with a lower success rate (especially for our approach) and more time needed to run the experiment.

D. Summary and Discussion

As shown in Table I, our approach reliably achieved good success rates in all simulated situations, whereas the baselines could not properly handle all of them. Our approach improves upon the *naïve baseline*, which has problems in S5 and S6 due to humans blocking access to buttons, humans standing too close to the door, or not letting humans exit, leading to mutual blockage. Compared to the *Veloso et al. baseline*, which has problems with S1 and S6 because no humans are available that can be asked for help, our approach did better as it does not depend on human help in this case. We could also find valid robot base poses for navigating in front of the button, inside, and in front of the elevator.

TABLE I
SUCCESS RATES IN SIMULATION (%), 20 EXPERIMENTS

Approach	S1	S2	S3	S4	S5	S6	S7	avg
Naïve	**95**	0	90	100	**15**	**35**	0.0	47
Veloso et al.	0	90	**95**	**85**	100	0	**85**	65
Ours	90	**95**	100	100	100	100	55	**91**

We found that asking humans to press a button when they already stand in front of it (S2) is always faster in our simulation than asking them to make space so that the robot can press the button on its own, as shown by the results of S2, S3, and S7 in Fig. 9. This suggests that our approach would be even faster if we opted for this strategy instead. In the real experiments, we saw that the robot could reliably position itself on the correct side of the button when humans were standing close, allowing the robot to press the button without colliding with the human.

Additionally, we measured the elevator entering time. Our robot usually needed less than 6 s between the door opening and the robot being inside the doors' light barrier, with some outliers being significantly slower. When the robot is too slow, it is not able to get into the elevator, and the doors close in front of it. However, collisions with the door can also occur in case the robot is usually fast enough, e. g.,, when its entering is hindered by humans. To allow for safe interactions with humans and the elevator in open-ended environments, the current premature safety measures need to be extended. In real robot experiments, we discovered that the button-pressing skill had a limited success rate. The button detection and pose estimation failed in less than 20 %, which can lead to the robot moving the fingers in the wrong direction. The pressing itself worked in around half the cases, but depended heavily on the calibration. Sometimes, the robot also did not identify that it had already reached its goal early enough, which eventually led to recalculations and oscillations between multiple goals.

V. CONCLUSION AND FUTURE WORK

In this work, we have presented a novel approach to combine autonomous elevator navigation with asking humans for help. Several experiments with varying complexity in simulation and on the humanoid robot ARMAR-7 demonstrate improved performance in diverse scenarios compared to purely autonomous or exclusively help-seeking approaches.

We identified several limitations for real-world deployment that require future investigation: To avoid collisions, we manually held elevator doors open since our robot currently lacks a strategy to safely enter the elevator. A potential solution could be to extend an arm toward the elevator door to keep it open during entry. If the door begins to close before the arm reaches the doorway, the robot can quickly retract it to prevent collisions. Another limitation is that unsuccessful button-pressing attempts require human intervention since the robot cannot currently verify button activation. Future work will integrate button state detection to enable verification of button activation. The work only considered simplified scenarios with limited human presence and assumed cooperative human behavior. Under these conditions, asking humans for help proved faster than letting the robot press the buttons on its own – especially when the human is already standing near the control panel. However, comprehensive user studies are required to assess the willingness of humans to help the robot, cooperation levels, and the overall user acceptance and satisfaction. Finally, extending the approach to more complex environments, such as crowded elevators, requires further research.

ACKNOWLEDGMENT

The authors thank Tobias Gröger, Corvin Navarro Ecker, Lennart Großkreutz, Marius Baden and Timo Weberruß for work on human-aware navigation, Peter Albrecht, Simon Scharpf and Leonie Leven for support with the experiments, and Bálint Máté for proxemic modelling.

REFERENCES

[1] M. Veloso, J. Biswas, B. Coltin, and S. Rosenthal, "CoBots: Robust Symbiotic Autonomous Mobile Service Robots," *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[2] S. Rosenthal and M. Veloso, "Mobile Robot Planning to Seek Help with Spatially-Situated Tasks," *AAAI Conference on Artificial Intelligence*, vol. 26, no. 1, pp. 2067–2073, 2021.

[3] M. Daza, D. Barrios-Aranibar, J. Diaz-Amado, Y. Cardinale, and J. Vilasboas, "An Approach of Social Navigation Based on Proxemics for Crowded Environments of Humans and Robots," *Micromachines*, vol. 12, no. 2, p. 193, 2021.

[4] F. Reister, M. Grotz, and T. Asfour, "Combining Navigation and Manipulation Costs for Time-Efficient Robot Placement in Mobile Manipulation Tasks," *Robotics and Automation Letters (RA-L)*, vol. 7, no. 4, pp. 9913–9920, 2022.

[5] J. Palacín, R. Bitriá, E. Rubies, and E. Clotet, "A Procedure for Taking a Remotely Controlled Elevator with an Autonomous Mobile Robot Based on 2D LIDAR," *Sensors*, vol. 23, no. 13, p. 6089, 2023.

[6] T. Kim, G. Kang, D. Lee, and D. H. Shim, "Development of an Indoor Delivery Mobile Robot for a Multi-Floor Environment," *IEEE Access*, vol. 12, pp. 45 202–45 215, 2024.

[7] D. Troniak, J. Sattar, A. Gupta, J. J. Little, W. Chan, E. Calisgan, E. Croft, and M. Van Der Loos, "Charlie Rides the Elevator – Integrating Vision, Navigation and Manipulation towards Multi-floor Robot Locomotion," in *International Conference on Computer and Robot Vision*, 2013, pp. 1–8.

[8] A. A. Abdulla, H. Liu, N. Stoll, and K. Thurow, "A New Robust Method for Mobile Robot Multifloor Navigation in Distributed Life Science Laboratories," *Journal of Control Science and Engineering*, vol. 2016, pp. 1–17, 2016.

[9] S. Jiang, W. Yao, M.-S. Wong, M. Hang, Z. Hong, E.-J. Kim, S.-H. Joo, and T.-Y. Kuc, "Automatic Elevator Button Localization Using a Combined Detecting and Tracking Framework for Multi-Story Navigation," *IEEE Access*, vol. 8, pp. 1118–1134, 2020.

[10] H. Zhang, W. Tao, J. Huang, and R. Zheng, "Development of An In-building Transport Robot for Autonomous Usage of Elevators," in *International Conference on Intelligence and Safety for Robotics (ISR)*, vol. 43, 2018, pp. 44–49.

[11] P.-Y. Yang, T.-H. Chang, Y.-H. Chang, and B.-F. Wu, "Intelligent Mobile Robot Controller Design for Hotel Room Service with Deep Learning Arm-Based Elevator Manipulator," in *International Conference on System Science and Engineering (ICSSE)*, 2018, pp. 1–6.

[12] D. Zhu, Y. Fang, Z. Min, D. Ho, and M. Q.-H. Meng, "OCR-RCNN: An Accurate and Efficient Framework for Elevator Button Recognition," *Transactions on Industrial Electronics*, vol. 69, no. 1, pp. 582–591, 2022.

[13] S. Manzoor, E.-J. Kim, G.-G. In, and T.-Y. Kuc, "Performance Evaluation of YOLOv3 and YOLOv4 Detectors on Elevator Button Dataset for Mobile Robot," in *International Conference on Control, Automation and Systems (ICCAS)*, 2021, pp. 890–893.

[14] J. Rios-Martinez, A. Spalanzani, and C. Laugier, "From Proxemics Theory to Socially-Aware Navigation: A Survey," *International Journal of Social Robotics*, vol. 7, no. 2, pp. 137–153, 2015.

[15] C. Rosmann, W. Feiten, T. Wosch, F. Hoffmann, and T. Bertram, "Efficient Trajectory Optimization Using a Sparse Model," in *European Conference on Mobile Robots*, 2013, pp. 138–143.

[16] J. Kim and C. Nam, "Room for Me?: Mobile Navigation for Entering a Confined Space Using Deep Reinforcement Learning," in *International Conference on Ubiquitous Robots (UR)*, 2023, pp. 1–5.

[17] J.-G. Kang, S.-Y. An, W.-S. Choi, and S.-Y. Oh, "Recognition and Path Planning Strategy for Autonomous Navigation in the Elevator Environment," *International Journal of Control, Automation and Systems*, vol. 8, no. 4, pp. 808–821, 2010.

[18] D. Zhu, T. Li, D. Ho, and T. Zhou, "A Novel OCR-RCNN for Elevator Button Recognition," in *International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[19] A. A. Abdulla, H. Liu, N. Stoll, and K. Thurow, "An Automated Elevator Management and Multi-Floor Estimation for Indoor Mobile Robot Transportation Based on a Pressure Sensor," in *International Conference on Mechatronics – Mechatronika (ME)*, 2016.

[20] J. Zhao, Y. Chen, and Y. Lou, "A Human-Aware Robotic System for Mobile Robot Navigating in Multi-Floor Building with Elevator," in *WRC Symposium on Advanced Robotics and Automation (WRC SARA)*, 2019, pp. 178–183.

[21] W.-t. Law, K.-s. Li, K.-w. Fan, T. Mo, and C.-k. Poon, "Friendly Elevator Co-rider: An HRI Approach for Robot-Elevator Interaction," in *International Conference on Human-Robot Interaction (HRI)*, 2022.

[22] G. Diego and T. K. O. Arras, "Please Do Not Disturb! Minimum Interference Coverage for Social Robots," in *International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 1968–1973.

[23] L. Scandolo and T. Fraichard, "An Anthropomorphic Navigation Scheme for Dynamic Scenarios," in *International Conference on Robotics and Automation (ICRA)*, 2011, pp. 809–814.

[24] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-Aware Robot Navigation: A Survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, 2013.

[25] E. T. Hall, *The Hidden Dimension*. Doubleday, 1966.

[26] S. Saunderson and G. Nejat, "How Robots Influence Humans: A Survey of Nonverbal Communication in Social Human–Robot Interaction," *International Journal of Social Robotics*, vol. 11, no. 4, pp. 575–608, 2019.

[27] G. Skantze, A. Hjalmarsson, and C. Oertel, "Exploring the Effects of Gaze and Pauses in Situated Human-Robot Interaction," in *Proc SIGDIAL 2013 Conf*, 2013.

[28] M. Shiomi, M. Kakio, and T. Miyashita, "Who Should Speak? Voice Cue Design for a Mobile Robot Riding in a Smart Elevator," in *International Conference on Robot and Human Interactive Communication (ROMAN)*, 2024, pp. 2023–2028.

[29] M. Iovino, E. Scukins, J. Styrud, P. Ögren, and C. Smith, "A Survey of Behavior Trees in Robotics and AI," *Robotics and Autonomous Systems*, vol. 154, p. 104096, 2022.

[30] F. Peller-Konrad, R. Kartmann, C. Dreher, A. Meixner, F. Reister, M. Grotz, and T. Asfour, "A Memory System of a Robot Cognitive Architecture and its Implementation in ArmarX," *Robotics and Autonomous Systems*, vol. 164, p. 104415, 2023.

[31] K. Luttgens, H. Deutsch, and N. Hamilton, *Kinesiology: Scientific Basis of Human Motion*. Brown & Benchmark, 1992.

[32] C. Gäbert, S. Kaden, and U. Thomas, "Generation of Human-like Arm Motions using Sampling-based Motion Planning," in *International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 2534–2541.

[33] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research*. Springer Berlin Heidelberg, 2011, pp. 3–19.