2025 International Conference on Advanced Computing and Intelligent Robotics Applications (ACIRA) | 979-8-3315-5742-3/25/$31.00 ©2025 IEEE | DOI: 10.1109/ACIRA67680.2025.11334740

"Elevator or Stairs" A Layered Reinforcement Learning Approach for Robot Multi-Floor Navigation in High-Traffic Scenarios

First Author: Shuanglong Luo
Guangdong Technology College. Zhaoqing. Guangdong. China
e-mail: luo591244113@163.com

Second Author: Qian Yu
Guangdong Technology College. Zhaoqing. Guangdong. China
e-mail: yuqianhuao123@163.com

Third Author: Chuang Zhu
Guangdong Technology College. Zhaoqing. Guangdong. China
e-mail: 18844217482@163.com

***Abstract* —Robots have major cross-floor navigation problems in high-rise buildings with large traffic. Conventional techniques are flawed in that they are not able to adequately measure dynamic expenses as well as multi-modal change costs. In this paper, a hierarchical reinforcement learning method derived on the complete visual perception concept is proposed, which enables an efficient decision-making process by having a manager-executor architectural design. The high-level strategy relies on a dual-stream visual network to combine semantic data in the environment to intelligently select the with either of the two possible movement options of either elevator or staircase, whereas the low-level strategy operates a sequential vision network (CNN-LM) to decode high-level instructions into continuous motion control commands, to achieve accurate movement and avoid obstacles. Using a non-linear rewarding system to calculate the dynamic costs and mode-switching punishment, in theory, average delivery time can be reduced by 15 to 25 percent and compliance of the task is increased to more than 95 percent. It is likely to perform much better than heuristic rules and available learning baselines. The suggested hierarchical decision model offers a new theoretical platform to address vertical logistics in complicated indoor setting where service robots operate that would offer valuable information to optimize autonomous decision-making abilities of robots in highly dynamic conditions.**

Keywords- hierarchical reinforcement learning; multimodal path planning; dynamic environments; robot navigation; visual navigation.

I. INTRODUCTION

The problem of the last-mile delivery is also managed with the broad use of logistics robots because of the rapid development of urban logistics. Nevertheless, central business districts (CBDs) and high-rise buildings are characterized by high traffic and thus efficient cross-floor delivery poses a big challenge to robots. Although there are physically challenged stair-climbing robots with capability of crossing physical obstacles, the most important issue in service robotics is still the ability to make proper intelligent decisions influenced by real-time conditions and calm down on whether to use the elevator or approach stairs (e.g. reduce time or energy use). Although the current literature mainly concentrates on navigation activities inside particular routes or individual planes, rather limited studies have examined the issue of multi-modal and dynamic environment integration to support the decision-making of delivery choices in high-rise. Much of the current HRL literature implicitly considers the case of taking the elevator and climbing steps as an isomorphic option between base actions in cross-floor problems, and only implicitly attempts to capture the underlying discrepancies in occupational choices in terms of their temporal cost, state transition processes and physical constraints. This leaves the strategies that are only adaptable in a very dynamic environment i.e. the queues in the elevator room.The paper will be constructed trying to develop a consonant model of decision making that can combine dynamic information on the environment with the state of the robot in performing intelligent mode selection.

II. . RELATED TECHNICAL RESEARCHSYSTEM MODEL

A. Classical Path Planning Algorithms

Classic path planning algorithms serve as the cornerstone for mobile robot navigation, primarily focusing on identifying collision-free routes from a starting point to a destination within a pre-established environmental model. These algorithms can generally be divided into two categories: graph-search-based methods, such as A*, and random-sampling-based techniques, like RRT.

The A* algorithm uses the functions of heuristics to perform effective searches in known grid or graph models with the purpose of making sure that the most efficient global paths will be found. It works very well in closed, instilled settings. By contrast, the RRT algorithm is a fast method to build spatial topologies by randomly sampling, demonstrating impressive flexibility to high-dimensional configuration spaces and complicated geometry. Nevertheless, this apparent limitation of classical algorithms is manifested when we move application situations out of controlled settings in the industrial domain to the disorganized real-life scenarios. The key problem lies in their supposition that the world stays the same, that it is observable completely and that the costs, or the costs associated to a path are constant or known, assumptions that are often in the true sense untestable in the dynamic, uncertain and the semantically interactive real world situations. In addition, classical algorithms are deficient in responding to state-dependent dynamic costs, and do not have the ability to

measure the costs of conducting heterogeneous transitions between behavioral modes.

Thus, although classical path planning algorithms are essential in local obstacle avoidance and global, which does not rely on dynamics, state-dependent and multi-modal costs, of the real-world problems, they have inherent limitations in the ability to deal with complex multi-dimensional problems [1].

B. Reinforcement Learning in Robot Navigation

Deep reinforcement learning (DRL) is a relatively new approach in autonomous navigation of a mobile robot. It has great powers of nonlinear greatly fitting and decision-making. There are also end-to-end navigation schemes, such as raw sensor values being mapped to low-level control commands. This produces more natural and strong-behaviour with no conventional mapping and localisation modules. As such, the different DRL algorithms have been extensively utilized on robot navigation with remarkable outcomes.

Nevertheless, DRL continues to struggle with such issues as the unattainable rewards and exploration in complicated and open-ended tasks. As a result of this challenge, Hierarchical Reinforcement Learning (HRL) was developed. HRL proposes the use of abstracted action spaces and time scales to subdivide tasks into high-level choices (what to do) and low-level implementation (how to do it). Literature studies have established that HRL is effective in enhancing policy generalization and learning efficiency [2-3].

However, there is a considerable disparity between multi-modal decision problems in the real world, e.g., cross-floor navigation. In this case, the robots will be required to strike a balance between various forms of behavior such as in taking the elevator, using stairs, or even waiting. All modes possess unique physical constraints, time costs as well as state dynamics. The prevailing studies of DRA navigation, however, are predominantly in the field of 2D plane navigation but with in just one mode. It does not have capacity to model the multi-modal decision in the vertical dimension [4]. The issue lies in the fact that the state space of reinforcement learning is usually not able to capture any essential dynamic data. This involves the length of queue in one of the elevators, when an elevator is idle or hiking, or the human traffic in stairways. This exclusion leaves robots unable to effectively consider these time sensitive variables, and this seriously constrains their applicability in the complex real world 3D context.

C. Elevator Scheduling and Stair-Climbing Robot Technology

In addressing the cross-floor movement, there has been a wide divide in the tech community between two different directions..

The initial one is the smart building route. This concept is concerned with environment modification to assist the robot, an excellent example of this concept is the elevator group control systems. Decisions are made in this case using the building system which factors in the optimum scheduling of the elevator to respond to the calls made by the robot. It is only passive users of this service who are the robots themselves.

The second option is the 'capability-based robot' solution. This solution focuses on enhancing the robot's physical capabilities. Crawler-type and bionic robots, for example, can climb stairs and overcome vertical obstacles autonomously, without external assistance. They make decisions based entirely on their own physical state and environmental perception.

Recently, research has tried to bridge the gap between these two separate paths by integrating their methods. Hybrid approaches like PRM-RL, for example, combine the large-scale structure of classical Probabilistic Roadmaps (PRM) with the local decision-making of DRL to handle long-range navigation [5]. While these hybrids show potential, their main weakness is their reliance on pre-built maps. The costs on these maps can’t adjust in real time to highly dynamic factors like pedestrian density. Because of this, they are not yet equipped to solve real-time, multi-modal decision-making problems.

The core principle of our paper is “decoupling and fusion.” “Decoupling” means separating long-term mode selection from short-term motion control, which makes learning simpler. “Fusion” uses a dual-stream visual network to blend a global understanding of the scene with important local details, providing rich semantic information for high-level decisions. This design is meant to build a single decision model that can adaptively learn the dynamic costs of an environment and make effective multi-modal trade-offs.

III. ESTABLISH AN MDP MODEL

To solve the problems discussed, this paper defines the delivery task using a Markov Decision Process (MDP) framework.

$$M=(S,A,P,R,\gamma). \quad (1)$$

State Space (S):The state space is an assembled have-way,s = (s rob, s task, s elev, s stair), which carries all the information needed to make a decision-making. It consists of four major components: (1) Robot state (s rob): This is a vector where the robot is physically described: the 2D position of the robot (x, y), the floor where it is, and the angle of rotation (θ). (2) Task state s_task: This is used to determine the goal and the urgency of the mission with the target_floor and its level of priority. State s of the elevator: In order to make decisions about the elevator, this state consists of the physically observed queue length an estimated wait time based on queueing theory and the current state of the elevator (e.g., idle, ascending, descending, doors open). (4) Staircase state s stairs: To navigate stairs we make use of cameras to approximate pedestrian density and traffic flow. This knowledge is essential when it comes to the safe flow of people in the congested zones [6].

Action Space (A): A hierarchical action space is used to make decision-making as simple as possible: The high-level actions A h are: {MODE ELEVATOR, MODE STAIR } and are chosen by the high-level policy as a result of which macro-moving modes are chosen. Low-level actions A l: { v, w} output of low level policy: These are continuous linear and angular velocity in-plane motion control.

Transfer Function (P): In actual practical settings, it is difficult to build accurate state transition models. Hence, this

method uses model-free reinforcement learning that uses neural networks that implicitly learn and approximate the transfer function via large volumes of robot-environment interactions.

Reward Function (R): A composite reward function is designed to guide the robot toward efficient and safe task completion.

R(s,a,s')=R_goal+R_time+R_safety+R_mode_penalt（2）

Where:

R_goal: Large positive reward when successfully reaching the target position; 0 otherwise.

R_time: Small negative reward per time step to penalize time consumption.

R_safety: Large negative reward upon collision.

R_mode_penalty: Additional negative reward per time step when stair-climbing mode is selected, simulating energy consumption and mechanical wear to encourage cautious decision-making.

IV. Hierarchical Reinforcement Learning Decision Method Based on Full Vision

This paper presents a purely visual HRL method designed for complex MDPs. By decoupling high-level semantic decisions from low-level motion control, our approach creates a system that is more adaptable, easier to interpret, and more efficient to learn..

A. Overall Architecture

Our method is built on a two-layer “manager-executor” architecture:

High-level strategy (manager) π_h: This component takes in a global visual representation of the scene. Its job is to decide on a high-level plan, such as “take the elevator” or “use the stairs.” It operates over long timeframes and is responsible for making the big-picture decisions.

Low-level strategy (executor) π_l: Receives local visual observations and high-level intent, outputs continuous control commands (v, w). It operates at a high decision frequency, responsible for achieving safe, smooth navigation and obstacle avoidance under the guidance of high-level intent.

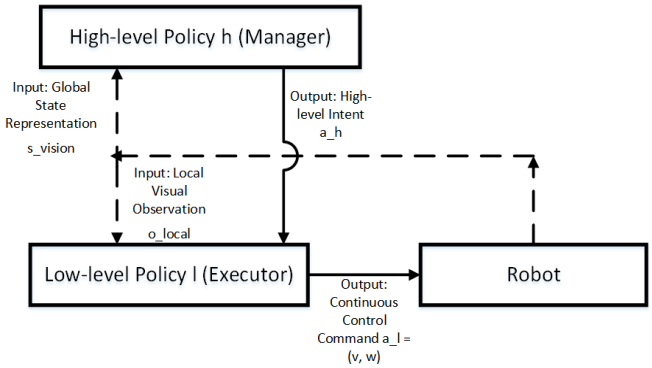

Figure 1. Two-layer architecture diagram of the overall method

As shown in Figure 1, the high-level policy operates at a lower frequency, receiving global visual information to perform macro-level pattern selection. This decision-making process considers long-term rewards. The low-level policy operates at a high frequency, receiving local visual observations and high-level commands to generate specific continuous motion control commands. This decoupled design reduces learning complexity and enhances the system's interpretability.

B. High-Level Policy: Visual Semantic Understanding Network

The core of the high-level policy is to distill visual information into compact state representations and perform pattern selection.

Network Architecture: The visual encoding network used is a two-stream one. A 224x224 RGB image is fed into a resnet-18 that has been trained to extract 512 dimensional global features into a stream into the global scene. The Local Attention Stream directly processes the cropped key regions by a lightweight CNN to get 128-dimensional local features. We are concatenating such inputs and sending them to a Multi-Layer Perceptron (MLP). An eventual Softmax layer will then result in the probability distribution of the pattern selection.

This model is trained with Proximal Policy Optimization (PPO). We set our main hyperparameters to the clipping coefficient (eps clip) = 0.2 and a discount factor (gamma) = 0.99 and a batch size = 64. The learning rate will begin with 3e-4 and linearly fade away. The choice of PPO compared to other algorithms such as SAC, or A2C, was mostly due to the theoretical characteristics that make it an appropriate algorithm when undertaking decision-making issues at the higher level. PPOs clipping mechanism is more stable during training compared to SAC when it is used with discrete high-level action space (e.g. mode selection). The experience reuse process of PPO is more efficient than A2C, so that the sample can be efficient which is needed by high-level learning of policy which demands long-term exploration. Its clipping mechanism provides stability to training, which is appropriate and effective in high-level semantic decision making that requires extended exploration.

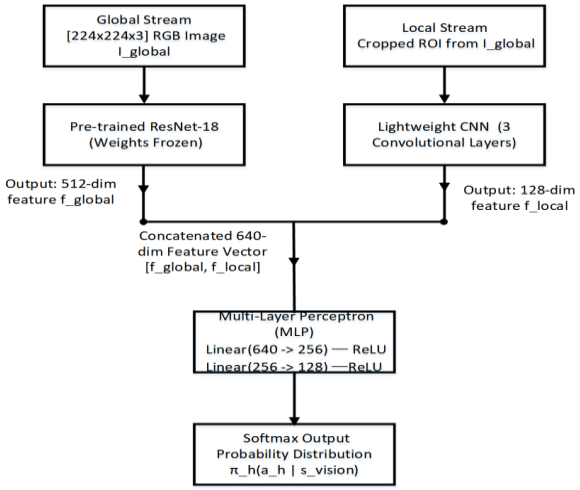

Figure 2. Dual-stream visual encoding network architecture for high-level strategies

As shown in Figure 2, the global scene stream utilizes ResNet-18 to extract macro-level semantic features f_global. The local attention stream processes key regions of interest (ROIs) through a lightweight CNN to extract refined features f_local. By concatenating f_global and f_local, the network simultaneously captures both the global environment and critical local details. The subsequent MLP is responsible for learning and making optimal pattern selections.

C. Low-Level Policy: Visual Motion Control Network

The low-level policy converts high-level commands into concrete motions, drawing inspiration from advanced methods in current visual navigation research that integrate temporal information [4].

Network Architecture: Input consists of four consecutive 128x128 local RGB frames (to perceive motion trends) and one-hot encoded high-level intentions. The stacked visual frames undergo encoding through four layers of CNN before being fed into an LSTM network for temporal dependency processing. The LSTM output is concatenated with the high-level intent, then passed through an MLP to generate continuous linear and angular velocities. These are mapped to the actual velocity range via a Tanh activation function.

Training Algorithm: The PPO algorithm is employed, whose Actor-Critic architecture effectively handles continuous action spaces while offering stable parameter tuning and convergence properties.

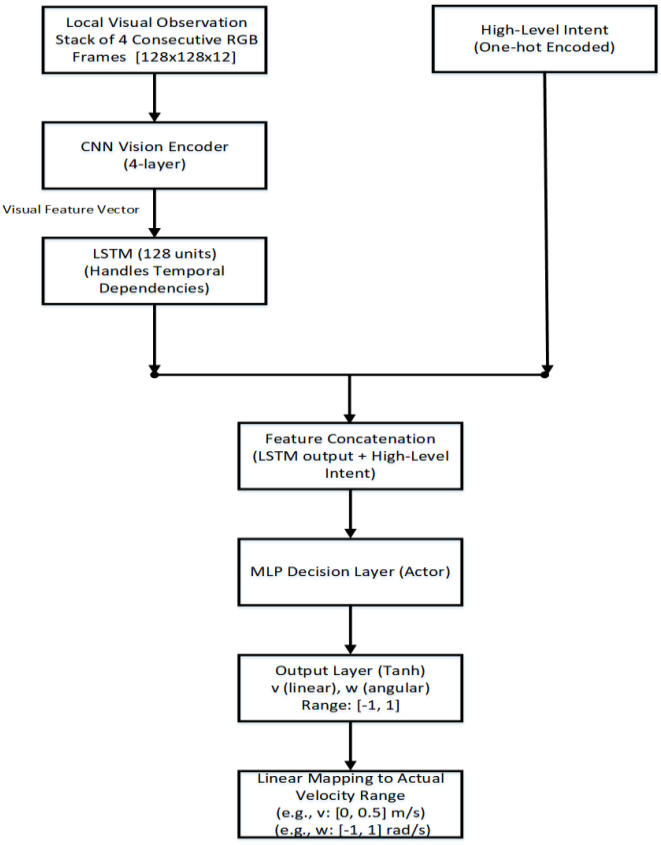

Figure 3. Visual-motor control network architecture for low-level strategies

The network uses four consecutive frame of the local RGB images to view the dynamic environment as indicated in Figure 3. Once CNN has been used to extract the temporal information, it is inputted to an LSTM to learn time dependencies. The result of the LSTM is fused with the high-level intent, and high-level intent is closely subjected to the where to go (high-level intent) and how to arrive safely (low-level perception). Lastly, an MLP transforms the combined information into explicit control it has been shown that strong obstacle avoidance and smooth path tracking is maintained with high-level goals guiding the control system.

V. Evaluation Metrics and Theoretical Expectation Analysis

A. Evaluation Metrics and Baseline Methods

To determine absolutely performance of methods, the following quantitative measurements have been set towards the comparison with the baseline methods.

Assessment Measures: (1) Average Delivery Time: Core efficiency measure. (2) Task Success Rate: Fraction of tasks as being completed within the time limit successfully. (3) Decision Pattern Distribution: Exemplifies the frequency with which elevators and staircases are chosen in different circumstances in order to test the strategy intelligence.

Baseline Methods:(1) Heuristic-Elevator: Chooses elevators always. (2) Heuristic-Stair: If there are stairs selected at all. (3) Rule-Based: Applies simple rules (e.g., choose elevator when it is peak time and target floor is above 3 rd floor). (4) Monolithic RL: End to end non hierarchy reinforcers training network that comes out with control messages directly as a result of visual output. PRM-RL: This is a hybrid advanced baseline that involves classical path planning and learning strategies [5].

B. Theoretical Expectations and Assumption

Assumption 1: The full-vision HRL method is expected to reduce average delivery time by 15%-25%.

Consider a task where a robot delivers documents across floors from Level 1 to Level 8, decomposed into three key decision scenarios:

Scenario A: Peak hours with elevator congestion. In this scenario, the vision system detects estimated_wait_time far exceeding the threshold.

Scenario B: Off-peak hours with available elevators. In this scenario, estimated_wait_time remains low.

Scenario C: Stair traffic is extremely high. In this scenario, stair_traffic_flow exceeds the safety threshold.

Assuming occurrence probabilities of 40% for Scenario A, 50% for Scenario B, and 10% for Scenario C, we estimate decision outcomes and expected durations for different baseline methods versus our HRL method in each scenario. As shown in Table 1:

TABLE I. TABLE 1 ESTIMATED TIME CONSUMPTION OF EACH METHOD IN DIFFERENT SCENARIOS

Scenario	Occurrence Probability	Baseline Method Average Duration	Full-Vision HRL Method Decision	HRL Method Estimated Duration
A: Elevator Congestion	40%	Heuristic_Elevator: Approx. 10 min; Heuristic-Stair: Approx. 5 min	Choose Stairs	Approx. 5 min
B: Elevator available	50%	Heuristic_Elevator: approx. 2 min; Heuristic-Stair: approx. 5 min	Choose elevator	Approx. 2 min
C: Stairs crowded	10%	Heuristic_Elevator: approx. 3 min; Heuristic-Stair: approx. 8 min	Choose elevator	Approx. 3 min

Through weighted calculations, the average duration of the heuristic baseline is approximately 5.3 minutes, while the fully visual HRL method takes about 3.3 minutes, representing a theoretical time savings of approximately 38%. Considering the perception and decision-making deviations of the model in real dynamic environments, the theoretical fully visual HRL method is expected to reduce the average delivery time by 15%-25%.

Assumption 2: The task success rate of the fully visual HRL method will exceed 95%.

Task failures primarily stem from timeouts and collisions/stalls. Our method theoretically addresses these issues:

Avoiding timeouts: HRL's dynamic decision-making mode fundamentally prevents the inevitable timeouts of heuristic methods in specific scenarios.

The risk of collision: Hierarchical structure allows lower-level strategies to devise robust obstacle avoidance learning that is not theoretically dependent on end-to-end models. Also the Rsafety element of the reward function is highly incentivizing as far as avoidance motivator goes.

On this basis, we can conclude that primary failure modes (timeout and collision) can be minimised by more than 80%. With a base success rate of about 90 in a test set-up, a general success rate of below 5 percent can be achieved and hence a theoretical enhancement of the task success rate to above 95.

VI. CONCLUSION AND OUTLOOK

The paper suggests a hierarchical reinforcement learning (HRL) to solve the problem of dynamic decision-making in the high-traffic environment across the floors of higher floors by robots moving in high-traffic settings. This method breaks down the complex task to have two layers that are: the high-level semantic decision-making level and the low-level motion control level. The top-level policy uses a dual-track visual network to integrate both the global and local information that allows them to switch between modes, which are the elevator and the staircase, intelligently. The low-end policy applies a sequential visual network in converting high-level commands into safe and smooth control actions.

This study proposes an entirely end-to-end vision-based methodology, which totally performs without the use of pre-compiled environment maps. In contrast to the hybrid approaches, including PRM-RL, which rely on pre-created maps, this solution acquires information regarding dynamic costs directly using raw visual inputs, and its ability to acquire them sharply contrasts with other methods. The essence of its innovation is that high level pattern selection and low level motion control are used in one High-Level-to-Low-Level (HRL) model, thus eliminating the problems of multimodal long-range navigation.

According to the theoretical analysis, this approach will be infinitely more effective than heuristic rules and non-hierarchical models in terms of such key metrics as the timeliness of delivery and success rate. This hierarchical design offers an effective, dependable as well as comprehensible resolution, bringing about new avenue to tackle the challenge of the last-mile vertical logistics robot delivery.

Although this method has theoretical benefits, it has a number of limitations: (1) The visual perception network needs to be tested to determine its resilience in a complex real-world situation including the difficult lighting conditions and occlusions. (2) Both the dual-stream visual network and the LSTM model are resource intensive and may therefore not be viable to apply in real-time to low-cost robots. (3) MDP model used fails to take into account complex real world interaction scenarios like the failure of elevators and scheduling multiple robots.

Future research will address: (1) Simulation versus reality: The research will see the gap between simulation and reality filled by conducting simulation experiments and applying them in high-rise buildings. (2) We plan to investigate methods such as knowledge distillation and model pruning to scale down computational power on embedded computing hardware. (3) The model can be further extended to multi-robot cases and cooperative strategies of navigation can be explored.

REFERENCES

[1] Zhang, H., et al. (2022). Socially-Aware Robot Navigation in Crowded Dynamic Environments Using Deep Reinforcement Learning. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

[2] Lazaric, A., & Ghavamzadeh, M. (2020). Foundations and Trends® in M achine Learning: Hierarchical Reinforcement Learning. Now Publishers.

[3] Chen, C., Luo, J., & Zhang, H. (2023). End-to-End Multimodal Navigati on for Urban Robots via Hierarchical Reinforcement Learning. IEEE Ro botics and Automation Letters, 8(6), 3456-3463.

[4] Kahn, G., Villaflor, A., Ding, B., Abbeel, P., & Levine, S. (2021). Uncer tainty-aware reinforcement learning for safe robot navigation in crowds. IEEE International Conference on Robotics and Automation (ICRA), 12 423-12430.

[5] Faust, A., Ramirez, O., Oslund, T., Francis, A., Chiang, H. T., & Fiser, M. (2022). PRM-RL: Long-range robotic navigation in real-world enviro nments using learned policies. IEEE Robotics and Automation Letters, 7 (2), 2360-2367.

[6] Riemer, M., et al. (2020). Learning to Navigate in Complex Environment s. International Conference on Learning Representations (ICLR).