DOI: 10.23977/avte.2026.080110
ISSN 2523-5834 Vol. 8 Num. 1

Teaching Reform and Practical Exploration of Robot Path Planning Programming Based on the Artificial Large Language Platform

Xiankun Lin[1,a,*], Linsen Liang[1,b]

[1]School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai, China

[a]linxk333@126.com, [b]233381595@st.usst.edu.cn

**Corresponding author*

Keywords: Artificial Intelligence; Robotics; Path Planning; Teaching Reform; Reinforcement Learning; ROS

Abstract: With the rapid advancement of artificial intelligence technology, robot path planning has shifted from traditional algorithm-driven approaches to data and model-driven paradigms. To address common challenges in university-level Robot Path Planning courses—such as overly abstract theoretical concepts, difficulties in programming implementation, and outdated algorithmic frameworks—this study proposes an AI-assisted teaching model incorporating a large language model (LLM) platform. Through a tiered curriculum architecture, a hybrid virtual-physical laboratory environment, and project-based instructional strategies, the proposed model seeks to equip students with proficiency in applying state-of-the-art techniques—from classical algorithms to deep reinforcement learning—to path planning challenges, fostering interdisciplinary mindsets and hands-on engineering competencies.

1. Introduction

With the in-depth advancement of the New Generation Artificial Intelligence Development Plan, intelligent robots have emerged as a key breakthrough in scientific and technological innovation. Robotic technology has transcended the conventional integration of traditional mechanical structures and classical control, and has achieved deep integration with key technologies including computer vision, natural language processing, and deep reinforcement learning. Path planning, as a core component of autonomous navigation for mobile robots, directly determines whether students can adapt to the development requirements of the future intelligent industry through its teaching quality. Meanwhile, generative technologies represented by large language models are reshaping the patterns of human-computer interaction and knowledge acquisition, providing new possibilities for the innovation of educational models.

However, prominent problems still exist in the current teaching of robot path planning courses in universities. First, the curriculum content lags behind technological development. Most courses take classic algorithms such as Dijkstra, A*, artificial potential field method and rapidly-exploring random trees (RRT) as the core, but provide insufficient coverage of modern application scenarios related to dynamic unknown environments, multi-robot collaboration and deep reinforcement learning. Cutting-edge content is mostly limited to postgraduate teaching, leading to an obvious knowledge gap at the undergraduate level. Second, there is a serious disconnection between theoretical teaching and engineering practice. The explanation of algorithms mostly stays at the level of simple simulation in MATLAB, and 2D grid-based path demonstration can hardly reflect the kinematic constraints, sensor noise and real-time requirements of real robots, resulting in weak engineering application capabilities of students[1]. Third, students lack competence in engineering programming and the application of intelligent tools. They have little practical experience in deploying algorithms in the robot operating system (ROS) environment and conducting GPU-based model training and tuning. They also lack intelligent learning approaches to carry out code debugging, concept understanding and scheme design with the support of large language model platforms.

Therefore, the integration of technologies related to large language models into robot path planning teaching is not only a demand for knowledge updating, but also a transformation of thinking modes. Technologies associated with large language models, especially reinforcement learning, provide a new approach for the shift from model-based planning to data-driven learning, enabling robots to self-evolve in the process of interacting with the environment. Moreover, large language model platforms further offer intelligent support for the whole teaching process: they serve as 24-hour teaching assistants for students, providing real-time guidance and code explanation in programming practices, and also act as instructional designers for teachers, assisting in generating differentiated teaching resources and personalized learning paths[2]. Thus, constructing a large language model-enabled teaching system for robot path planning is of great practical significance to cultivate students' interdisciplinary systematic thinking and equip them with full-stack capabilities covering problem definition, model selection, code writing, algorithm debugging and physical deployment.

2. Pedagogical Philosophy and Learning Outcome Design

This curriculum reform follows the philosophy of student-centered and outcome-based education. In conjunction with the standards for engineering education accreditation, the teaching objectives are deconstructed into three dimensions: knowledge, capability and literacy. Meanwhile, the LLM platform is employed as an intelligent supporting tool that runs through the above three dimensions, as shown in Figure 1[3].

(1) In terms of the knowledge dimension, students are required to master the modeling methods of simulation environments for path planning, including the map construction principles of the grid method, geometric method and topological method, and understand the update mechanism of occupancy grid maps. Students are expected to fully command classical planning theories, deeply understand the completeness and optimality of graph search algorithms, and grasp the principle of probabilistic completeness of sampling-based algorithms in continuous spaces. Through conversational inquiries, students can ask the LLM to explain abstract theoretical concepts with diverse metaphorical examples, generate comparison tables of theoretical concepts, or provide detailed mathematical derivation procedures of algorithms, so as to deepen their comprehension of theoretical knowledge.

(2) In the dimension of capabilities, the course requires students to possess algorithm implementation capabilities: they should not only proficiently use Python/C++ to implement core algorithms ranging from A* to Deep Q-Network (DQN), but also conduct complexity analysis and optimization of these algorithms. Additionally, students are expected to have simulation

development capabilities, including the ability to build a ROS workspace, construct path planning simulation environments using Gazebo or CoppeliaSim, and realize robot model control and sensor data reading. The course also cultivates students' model debugging capabilities, enabling them to master code debugging skills in deep learning training and propose improvement solutions for the problem of training non-convergence. Students can input error messages or abnormal training curves into the LLM to obtain debugging suggestions. Moreover, the LLM can generate code templates, explain API usage, and provide algorithm pseudocode. This lowers the threshold for programming entry and accelerates the transformation of course knowledge from theoretical understanding to practical application.

(3) In terms of student literacy, the course aims to cultivate students’ systematic thinking by regarding perception, planning and control as an organic integrated system, and helping them understand the positioning of the input and output of the path planning module within the system. The course incorporates content to foster students’ engineering ethics and safety awareness. The safety-first principle is introduced into path planning, enabling students to understand that robot obstacle avoidance is not merely an algorithmic issue but also a safety concern.

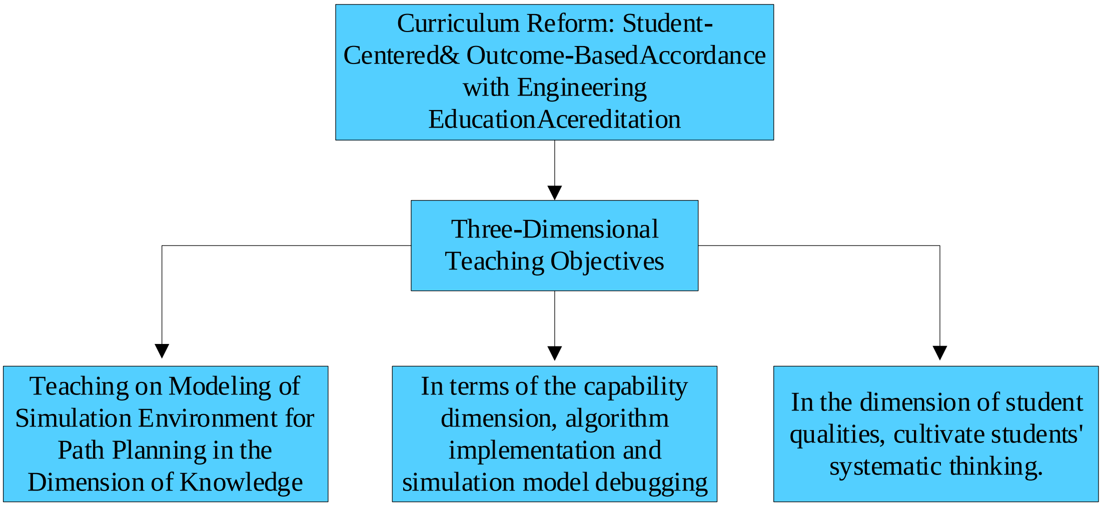
Curriculum Reform: Student-Centered& Outcome-Based Accordance with Engineering EducationAcereditation
Three-Dimensional Teaching Objectives
Teaching on Modeling of Simulation Environment for Path Planning in the Dimension of Knowledge
In terms of the capability dimension, algorithm implementation and simulation model debugging
In the dimension of student qualities, cultivate students' systematic thinking.

Fig. 1 Three-Dimensional Teaching Objectives of Robot Path Planning Course.

3. Curriculum Architecture

To achieve the above teaching objectives, we reconstruct the traditional teaching content and design four progressive modules, forming a closed teaching loop from theory to cutting-edge applications. Large language model-related technologies are deeply integrated into each module, and a large language model platform is introduced to provide intelligent support.

3.1. Module of Environmental Perception and Simulation Environment Modeling for Path Planning

This module covers coordinate transformation and localization, including rigid body pose description, rotation matrices, translation vectors, odometry models and cumulative errors. The part of simulation map environment construction focuses on the principle of occupancy grid maps, which discretize the continuous physical world into grids, with each grid value representing the probability of an obstacle. This course illustrates the application of image segmentation methods based on convolutional neural networks in obstacle category recognition. Taking the pre-trained Mask R-CNN model as an example, it compares the differences between traditional occupancy grid maps and semantic maps [4].

Semantic maps can assign differentiated costs to obstacles of different categories, thus laying a foundation for the realization of differentiated path planning.

Students can leverage LLM to learn the mathematical principles of occupancy grid maps, with the models providing code examples of Bayesian updates for different sensor models. In terms of experimental content, the course takes lidar scan data as input and requires students to write Python programs with the assistance of LLM to complete the update operation of a 20×20 grid map. Throughout the experimental process, LLM can not only help students gain a deep understanding of algorithm details but also guide them to reflect on the trade-off between grid map resolution and computational efficiency.

3.2. Classical Path Planning Module

This module primarily covers graph search algorithms and sampling-based algorithms. For Dijkstra's algorithm, the focus is on explaining its breadth-first search characteristics and the principles of solving the shortest path in graphs with non-negative edge weights. For the A* algorithm, key points are elaborated on the design principles of heuristic functions: the differences between Manhattan distance and Euclidean distance in diagonal motion models are compared, and the guiding effect of heuristic functions on search directions and the optimization effect on node expansion are analyzed. The section on sampling-based algorithms introduces the random sampling mechanism, probabilistic completeness, and non-optimality of the Rapidly-exploring Random Tree (RRT) algorithm in continuous high-dimensional spaces, and demonstrates the expansion process of its tree structure through animations. For RRT, the operations of parent node reselection and rewiring are explained, and the implementation principles of its asymptotic optimality are illustrated. The course adopts the "maze treasure hunt" as a teaching case, with the assistance of the LLM for practical exercises, enabling students to implement the A* algorithm on 2D grid maps. Students can leverage LLM to understand the algorithm flow: parse core steps through flowcharts, and compare the search differences between Dijkstra's algorithm and the A* algorithm. Based on the heuristic function weight adjustment schemes provided by LLM, students can observe changes in search efficiency and path optimality, and reflect on the trade-off between the two. Additionally, LLM can assist in generating visualization code to intuitively present the algorithm search process.

3.3. LLM-based Intelligent Planning Module

This module is the core of the course, realizing the transition from deterministic planning to learning-based planning, and is divided into three teaching units. The first unit focuses on the principle, limitations and improved methods of the artificial potential field method. In teaching, the attractive and repulsive potential functions are reviewed, the resultant force formula is derived, and the problems of oscillation in narrow channels and local minimum traps in the artificial potential field method are pointed out. The genetic algorithm is introduced to optimize the parameters of the potential field method, and automatic optimization of robust parameters is realized through parameter encoding and fitness function evaluation. LLM can assist students in understanding the selection, crossover, and mutation operations of genetic algorithms, provide fitness function design templates and parameter encoding suggestions, and guide students to compare the similarities and differences between genetic algorithms and classical optimization methods. The second unit serves as an introduction to reinforcement learning. Taking the grid world as an example, it completes the modeling of the Markov Decision Process (MDP), derives the Bellman equation, explains the state value function and action value function, and focuses on introducing the model free off policy Q-Learning algorithm and its core update formula[5]. When students implement Q-Learning in Python, they can leverage LLM to understand the ε greedy strategy and exploration rate decay strategy, obtain code for visualizing learning curves, analyze the impact of different reward functions on learning outcomes, and deepen their understanding of the balance between exploration and exploitation. The third unit focuses on deep reinforcement learning, elaborating on the evolutionary logic from Q-Learning to DQN. For continuous or high-dimensional state spaces, deep neural networks are introduced as function approximators, with emphasis on two key technologies: experience replay and target network[6]. In teaching, CartPole from OpenAI Gym is used as an introductory case, and the method is transferred to a custom robot simulation environment to achieve end-to-end path planning. LLM can assist students in understanding the design of neural network structures, generate DQN code frameworks for PyTorch or TensorFlow, parse the implementation logic of the experience replay buffer, and provide suggestions for loss function oscillation handling and hyperparameter tuning.

3.4. Multi-Agent Coordination and Dynamic Collision Avoidance

This section introduces the concept of multi-agent path planning and explains the principle of the Conflict-Based Search (CBS) algorithm. The algorithm adopts a hierarchical architecture: the high level searches for and identifies conflicts among agents, and the low level plans paths independently for each agent. In the AI-integrated direction of multi-agent path planning, multi-agent reinforcement learning methods are introduced. The core framework of centralized training with decentralized execution is highlighted, and the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm is taken as an example to illustrate its design logic: the critic network of each agent obtains global state information, while the actor network only relies on local observation data. In the LLM-assisted part, students can use LLM to compare the strengths and weaknesses of the traditional CBS algorithm and the MADDPG algorithm, ask LLM to provide guidelines for building a multi-agent simulation environment, or interpret the design philosophy of "centralized training with decentralized execution" and its application value in path planning scenarios. In addition, LLM can guide students to think about the scalability of the algorithm, such as the design of cooperative strategies for multiple agents under limited communication bandwidth.

4. Simulation Practice Platform and Environment Construction

To achieve the above-mentioned teaching objectives, this course reconstructs the traditional teaching content of path planning into four progressive modules, and deeply integrates the LLM platform as an intelligent auxiliary tool throughout the entire process. Students can invoke LLM at any time during the experimental process to obtain multi-faceted support such as code debugging, concept analysis, and experimental design, truly realizing a practical teaching model of human-AI collaboration.

In the basic module of environmental perception and modeling, the course not only teaches the construction principles of occupancy grid maps but also introduces concepts related to semantic maps based on deep learning. Traditional occupancy grid maps can only characterize the existence state of obstacles, failing to distinguish between different obstacle categories, and thus can only support simplified obstacle avoidance strategies. For example, if a robot adopts the same obstacle avoidance strategy for pedestrians and chairs, it may not only disturb pedestrians but also reduce operational efficiency due to overly conservative strategies. With the help of pre-trained convolutional neural networks (CNNs) such as Mask R-CNN, students can complete object classification in camera images and annotate semantic information in grid maps. LLM can assist students in understanding the working mechanism of Mask R-CNN, explain the implementation method of mapping image segmentation results to the grid coordinate system, and generate communication examples between ROS nodes and Python scripts. This teaching process helps students master the integration methods of computer vision and robot perception, and intuitively understand the core logic of how artificial intelligence drives the transformation of environmental modeling from binary modeling to semantic modeling.

In the classical path planning algorithms module, the course retains content on classic algorithms such as Dijkstra, A*, and RRT, while introducing LLMs to optimize and compensate for the deficiencies of traditional algorithms. Taking the artificial potential field method as an example, this algorithm features simple computation but has an inherent flaw of local minima. In this course, genetic algorithms are introduced to realize automatic optimization of the parameters of the artificial potential field method. Students are required to first master the principles and flaws of the artificial potential field method, then learn how to encode parameters into chromosomes, and design a fitness function that includes evaluation indicators such as path length, whether the algorithm falls into local minima, and the minimum distance to obstacles. After multiple generations of evolutionary iteration, the genetic algorithm can automatically search for the optimal parameters that maximize the success rate of path planning. LLMs can assist students in understanding the convergence analysis of genetic algorithms, generate visualization code for the parameter search process, and explain the reasons why specific parameter combinations lead to local minima. Through this process, students can personally experience how LLM optimization technology overcomes the blindness of manual parameter tuning, thereby stimulating their interest in exploring the field of integration between LLM platforms and path planning.

In the intelligent path planning module, the course incorporates an LLM-based intelligent planning section. Starting from the theoretical framework of Markov Decision Processes (MDPs), it gradually introduces the Q-Learning and DQN algorithms. LLM assistance runs through the entire learning and practice of the module. When implementing the Q-Learning algorithm, students can consult the LLM about code logic at any time. When non-convergence occurs during DQN training, they can submit loss curves and code snippets to the LLM for debugging suggestions. When comparing Q-Learning with the A* algorithm, students can use the LLM to generate comparative analysis tables, deepening their understanding of the paradigm differences between the two types of algorithms. This practical mode of human-LLM collaboration enables students to not only master algorithm implementation methods but also learn to solve complex engineering problems with the assistance of large language models.

In the multi-machine collaboration and dynamic obstacle avoidance module, the course introduces concepts related to multi-agent reinforcement learning. LLMs can provide students with the latest research advances in the field of multi-agent reinforcement learning, generate simplified implementation code for MADDPG, and explain the applicable scenarios of different multi-agent collaboration strategies. LLMs can also guide students to reflect on communication constraints and scalability issues in real multi-robot systems.

5. Design of Project-Based Learning Scenarios

To integrate scattered knowledge points into systematic practical abilities, the course designs a semester-long comprehensive project titled Intelligent Warehouse Robot, which simulates the intelligent handling operations of modern logistics warehouses. The project is implemented in three phases, with large language models serving as full-cycle intelligent assistants to support the entire project.

In the first five weeks, practices related to static environment mapping and path planning are carried out. Students work in groups to build a warehouse scene including shelves, charging piles and packing stations in the Gazebo simulation environment. They teleoperate the robot to traverse the environment, and use LiDAR data together with the Gmapping algorithm to complete the construction of an occupancy grid map, experiencing the SLAM process personally and understanding the uncertainty and cumulative errors in mapping. Then the A* algorithm is adopted for path planning, and point-to-point navigation of the robot is realized combined with a PID controller. The teacher temporarily adds static obstacles on the path to invalidate the original plan, which intuitively shows the limitations of traditional planning methods. LLMs can guide students to analyze why the A* algorithm cannot deal with unknown obstacles and discuss dynamic replanning schemes, so as to cultivate their ability of problem analysis and independent inquiry.

From the 6th to the 10th week, the course focuses on dynamic obstacle avoidance and intelligent decision-making. Mobile robots are introduced as dynamic obstacles in the warehouse environment. Abandoning static planning methods, students use Q-Learning to realize autonomous obstacle avoidance of the robot. By designing a state space centered on discretized LiDAR data and the relative orientation to the target, as well as a reward function that encourages approaching the target and penalizes collisions, the robot completes strategy learning through trial and error. With training iterations, the robot gradually acquires stable dynamic obstacle avoidance behavior. LLMs can provide support for state space design, reward function construction and training parameter optimization, helping students understand the adaptive advantages of data-driven methods.

From the 11th to the 16th week, the practice of AI optimization and physical logic integration is promoted. Students upgrade Q-Learning to DQN to process the high-dimensional state space composed of continuous LiDAR data, and expand continuous actions combined with the non-holonomic kinematic constraints of the robot to improve path smoothness. This phase requires the completion of neural network construction, experience replay buffer design and hyperparameter tuning, with a significant increase in technical challenges. LLMs can provide network structure analysis, core code implementation and optimization schemes for training non-convergence problems.

Given that traditional final assessments cannot truly reflect students' engineering practice abilities, the course has established a diversified evaluation system. Process evaluation accounts for 40% of the total grade, among which in-class quizzes account for 10%, focusing on the examination of algorithmic principles, and experimental reports account for 30%, which are required to include debugging logs and LLM application records. Outcome evaluation accounts for 20%, with the final examination assessing conceptual discrimination and system design abilities. Project incremental evaluation accounts for 40%, which evaluates robot navigation performance through defense and on-site demonstrations. Innovation bonus points are offered to encourage students to carry out algorithm improvement and scheme innovation with the assistance of large language models.

6. Conclusion

The teaching reform of programming involving LLMs and robot path planning is not a simple superposition of LLM algorithms onto robot courses, but a profound reconstruction of teaching methods. By integrating deterministic classical algorithms and probabilistic models, and introducing the LLM platform as a full-process intelligent assistant, students can establish a complete methodology for solving complex engineering problems: seeking optimal solutions based on rules in known environments, pursuing approximate optimal solutions via data in unknown environments, and obtaining real-time support.

In the future, with the development of large language models and multimodal large models, we can further explore their in-depth applications in robot task planning, elevating path planning from low-level motion control to high-level cognitive decision-making. Meanwhile, the value of LLMs in teaching evaluation can be exploited to diagnose knowledge blind spots based on students' interaction records and realize personalized teaching. In addition, LLMs can be applied to personalized exercise generation, interview simulation, and teachers' teaching reflection and curriculum iteration, which will become the core directions of the next stage of teaching reform.

Acknowledgements

This work was supported by Teacher Teaching Development Research Project of University of Shanghai for Science and Technology (CFTD2025YB10) and Demonstration Course Project of Integrating Ideological and Political Education into Courses of University of Shanghai for Science and Technology (KCSZ202503).

References

[1] Bhuvaneswari, M., Elango, C., Jagadeeshwaran, S., & Riyaz, J. S. M. (2025). Design and fabrication of robotic operating system based autonomous mobile robot for education and material handling purpose. AIP Conference Proceedings, 3204(1), Article 030019. https://doi.org/10.1063/5.0208809

[2] Xu, L., & Zhang, W. (2025). Survey on path planning based on deep reinforcement learning. In Proceedings of 2025 2nd International Conference on Machine Learning and Intelligent Computing (Proceedings of Machine Learning Research, Vol. 278, pp. 685–695). PMLR. https://proceedings.mlr.press/v278/xu25a.html

[3] Belpaeme, T., Kennedy, J., Ramachandran, A., Scassellati, B., & Tanaka, F. (2018). Social robots for education: A review. Science Robotics, 3(21), eaat5954. https://doi.org/10.1126/scirobotics.aat5954

[4] Gupta, S., Tolani, V., Davidson, J., Levine, S., Sukthankar, R., & Malik, J. (2020). Cognitive mapping and planning for visual navigation. International Journal of Computer Vision, 128(5), 1311–1330. https://doi.org/10.1007/s11263-019-01236-7

[5] Love, R., Law, E., Cohen, P. R., & Kulic, D. (2023). Adapting a teachable robot's dialog responses using reinforcement learning in teaching conversation. In 2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN) (pp. 2541–2548). IEEE. https://doi.org/10.1109/RO-MAN57019.2023.10309477

[6] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529–533. https://doi.org/10.1038/nature14236