

Reactive Path Planning for UAVs in Static Environments Using Adaptive Block Coordinate Descent Logistic Regression

Venkatasivarambabu Pamarthi

Koneru Lakshmaiah Educational Foundation

Khyathisree Yarra

SRM University-AP

Prasanthi Boyapati

`prasanthi.b@srmap.edu.in`

SRM University-AP

Ravi Sankar Puppala

Amrita School of Computing, Amrita Vishwa Vidyapeetham

Thulasi Bikku

Amrita School of Computing, Amrita Vishwa Vidyapeetham

Ramesh Reddy Donapati

VNR Vignana Jyothi Institute of Engineering and Technology

Article

Keywords: Unmanned Aerial Vehicles, Machine Learning, Linear Regression, Logistic Regression, Stochastic Gradient Descent Optimization, Block Coordinate Descent

Posted Date: October 9th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10992607/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Reactive Path Planning for UAVs in Static Environments Using Adaptive Block Coordinate Descent Logistic Regression

Venkatasivarambabu Pamarthi¹, Khyathisree Yarra²,
Prasanthi Boyapati^{3*}, Ravi Sankar Puppala⁴, Thulasi Bikku⁵,
Ramesh Reddy Donapati⁶

¹Department of Computer Science Engineering, Koneru Lakshmaiah Educational Foundation, Vaddeswaram, Guntur, 522 501, Andhra Pradesh, India.

^{2,3}Department of Computer Science and Engineering, SRM University-AP, Amaravati, 522 240, Andhra Pradesh, India.

^{4,5}Department of Computer Science and Engineering, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Amaravati, 522 503, Andhra Pradesh, India.

⁶Department of ECE, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, 500 118, Telangana, India.

*Corresponding author(s). E-mail(s): prasanthi.b@srmap.edu.in;
Contributing authors: sivarambabu424@gmail.com;
khyathisree.yarra@gmail.com; p.ravisankar@av.amrita.edu;
b.thulasi@av.amrita.edu; rameshreddy_d@vnrvjiet.in;

Abstract

Unmanned aerial vehicles (UAVs) require efficient and reliable path planning mechanisms for safe navigation in obstacle-constrained environments. This study presents an Adaptive Block Coordinate Descent Logistic Regression (ABCD-LoR) model for reactive path planning in static environments using low-cost sensor inputs. The proposed approach formulates obstacle avoidance as a three-class classification problem (no turn, left turn, right turn), where navigation decisions are made sequentially based on local observations rather than pre-computed global trajectories. The model parameters are optimized using an adaptive block coordinate descent strategy, which improves convergence stability

and reduces sensitivity to noisy updates compared to stochastic gradient-based methods. Performance is evaluated using statistical model fit measures, including Pseudo R^2 , Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), log-likelihood, and likelihood ratio tests, along with classification metrics and ROC analysis. Simulation experiments are conducted in dense static environments under varying speed conditions. The proposed method is compared with classical path planning approaches, including A*, Vector Field Histogram (VFH), Fuzzy Logic Controller (FLC), Dynamic Window Approach (DWA), Artificial Potential Field (APF), and ASGD-based logistic regression. Results indicate that ABCD-LoR achieves reduced path length and lower computational time while maintaining competitive predictive performance. The study is based on a simulated dataset generated from infrared and ultrasonic sensor configurations. While the results demonstrate the feasibility of the approach for real-time reactive navigation, the limited dataset size and absence of sensor noise highlight the need for further validation in real-world and dynamic environments.

Keywords: Unmanned Aerial Vehicles, Machine Learning, Linear Regression, Logistic Regression, Stochastic Gradient Descent Optimization, Block Coordinate Descent

1 Introduction

The use of UAVs is seen in path planning for many purposes, such as logistics, warehousing, and manufacturing. One of the major characteristics of autonomous vehicles is their state-of-the-art sensory abilities. These vehicles are fitted with many sensors such as cameras, LIDAR, and proximity sensors, and as a result, they have the ability to sense their environment and detect the obstacles in order to avoid collision. Moreover, these UAVs have the capability of navigating in diverse environmental conditions such as dense and cluttered. In recent years, UAVs increasingly employ deep learning-based perception systems for robust navigation and obstacle avoidance [1]. These new technologies allow them to learn from their experiences and continuously improve their performance over time [2]. By applying any adaptive model, UAVs can plan their routes more efficiently, make quick decisions, and even work together with other Aerial vehicles to do challenging jobs [3] [4] [5]. For UAVs, a path planning operation is one of the essential requirements for UAV for safely navigating to avoid obstacles in a particular type of environment. A path planning algorithm has a critical task in guiding an autonomous vehicle from its initial point to its destination point by determining a feasible route. The primary objective of this algorithm is to generate a feasible path while considering computational efficiency required for the UAV to reach its goal while ensuring the feasibility of the path [6]. Ultrasonic (US), InfraRed (IR), and LIDAR sensors can be used in autonomous vehicles, and they provide information about nearby obstacles in real time. For ensuring safer navigation, some drones utilize mapping algorithms for constructing the environment [7]. As a result, the UAV can safely navigate by considering the different locations of obstacles, obtained from the sensors. making decisions related to path planning operations, such as graph-based methods such as Dijkstra's [8] and A*, sampling-based approaches such

as Rapidly exploring Random Trees (RRT) [4] and Probabilistic Roadmaps (PRM), or optimization strategies that compute the cost functions in determining the feasibility of the path. Learning-based algorithms possess the ability to learn patterns from data by understanding the relationship between the inputs of sensors and the presence of obstacles. While training the algorithm, the internal parameters (coefficients) of the models are updated at each iteration for enhancing the performance [2, 9]. Some systems fuse sensor fusion from odometry, GPS, and vision systems improves environmental understanding [10]. For improving the performance and to respond effectively to changing conditions of UAV, Reinforcement Learning (RL) techniques are applied for optimizing the control policies based on feedback and rewards [2, 9].

UAVs are often used in industrial applications where they are operated in pre-defined environments. With the advancement of the most current AI technology, Autonomous Vehicles can nowadays perform a wide range of tasks in an unfamiliar environment. Drones are used in military applications, including logistics and transportation. As a part of the surveillance systems, UAVs regularly visit predefined checkpoints within fixed boundaries. In recent years, UAVs have become more widely used across applications. For the above-mentioned applications of the UAV, various complex, vast, unstructured, and dynamic environments are present in real-world scenarios, navigating which pose a challenging task [11]. The UAV must navigate around obstacles of varying sizes and shapes. Because of inconsistencies in sensor data due to the data acquisition process of the sensor, a risk of colliding with obstacles remains an important challenge. Therefore, a model can be constructed to address these challenges. However, it becomes more challenging when the environment is complex, vast, and unknown. State estimation is the procedure to determine the estimate of the internal state by combining the mathematical model with input/output data measurements [12]. Solving many energy systems and control issues requires knowledge of the status of the energy system. Sensor placement depends on system requirements and constraints according to the requirement. It is not feasible to deploy sensors across the entire system in most applications, as it might be costly and pose challenges in managing it effectively [8, 13, 14]. Despite extensive research in UAV path planning, existing approaches either rely on global map knowledge or incur high computational cost in real-time applications. Reactive methods, while efficient, often lack statistical validation and interpretability. Furthermore, limited attention has been given to lightweight learning-based models that can operate under strict computational constraints using minimal sensor input.

Contributions of this work:

- Formulation of static-obstacle avoidance as a three-class logistic regression task using inexpensive IR + US sensors.
- Introduction of Adaptive Block Coordinate Descent as the optimizer for logistic weights, addressing outlier sensitivity and heteroscedasticity in comparison with stochastic gradient descent.
- Extensive simulation comparison in three realistic dense layouts at low/medium/high speeds, quantifying path length + CPU time against several established methods.

- Statistical model evaluation (Pseudo R^2 , AIC, BIC, LL ratio test) + classification performance (accuracy, precision, recall, AUC) missing in many reactive planners.

Despite the success of deep learning-based approaches, their high computational cost and lack of interpretability limit deployment on resource-constrained UAV platforms [15, 16]. Logistic regression is chosen over deep networks because it offers interpretability, extremely low inference latency (≈ 1 ms), and sufficient expressiveness for binary/multi-class turn prediction from low-dimensional sensor input. The main objective is to develop recent adaptive models for achieving faster and accurate decision capabilities of the UAV in addressing the path planning operation by avoiding static obstacles. Logistic Regression based models may be developed for the challenging task of path planning in different complex environments. Moreover, the distance of the generated path and the associated time need to be calculated and compared with other state-of-the-art path planning techniques. Various path planning algorithms have been applied for finding the optimal path in a static environment. However, the effectiveness of many previously proposed approaches in complex, dynamic, and partially unknown environments over long time periods has not been fully investigated. So far, limited research has addressed path selection in changing environments. In this dissertation, adaptive models have been proposed for path planning in avoiding static obstacles in dense and cluttered environments. The proposed approach formulates UAV navigation as a sequence of local decision-making steps, where each decision is derived from current sensor observations. Unlike global path planning methods, the model does not compute a complete trajectory in advance but determines the UAV motion incrementally at each time step.

The remainder of this paper is organized as follows. Section 2 reviews existing path planning approaches. Section 3 presents the proposed ABCD-LoR model and system architecture. Section 4 discusses experimental results and comparisons. Section 5 concludes the paper and outlines future work.

2 Literature Survey

This chapter contains a different literature survey on different widely used approaches based on adaptive models related to path planning in the presence of obstacle by considering different types of environments.

Gul et al. [17] proposed a hybrid PSO–GWO optimization framework for multi-objective path planning, focusing on global optimality and convergence efficiency. Through the weight estimation of the UAV configuration, an obstacle avoidance system has been designed. The fuzzy approach where the outputs of the fuzzy controller produce the signal where the steering motors are controlled. The proposed controller overcomes the limitation on the different range accuracies regarding avoiding obstacles. The proposed a fuzzy based controller which is developed on the general perception concept. The autonomous vehicle navigates through the environment by gathering spot information from ultrasonic sensors. Chen et al. [18] introduced a clustering-based coverage path planning method for heterogeneous UAVs, improving task allocation and spatial efficiency. Three linguistic rules are developed for controlling the direction of autonomous Vehicle. Developed an intelligent motion planning algorithm by

implementing fuzzy logic rules. Fuzzy rules are constructed by combining repelling and attracting influences with respect to the distance and angle values of the obstacle. Fuzzy logic-based obstacle avoidance methods utilize rule-based inference by combining distance and orientation information from sensors to generate navigation decisions in uncertain environments [17, 18].

The paper by Adil et al. [19] gave an extensive review of the use of UAVs in IoT applications, focusing on the quality-of-service issues and the need for communication aware path planning. The RNA BP network has been suggested as a solution for responding to situations involving low fault tolerant control systems. Neural-fuzzy hybrid learning methods have been tried out for UAV navigation applications [19]. Zhang et al. [20] proposed a bionic dynamic path planning algorithm that integrates deep neural network optimization with bio-inspired hawk-eye vision to jointly achieve obstacle avoidance and target tracking in micro-UAVs. Peng et al. [21] studied joint path planning and computation offloading optimization in UAV-enabled mobile edge computing systems. The scenarios explored involve the use of sensors to gather data from the environment, and this data was utilized by artificial intelligence-based models to make reactive decisions. These algorithms use an optimal control approach to address the path planning task. Lin et al. [22] proposed an improved artificial bee colony algorithm for UAV path planning, enhancing exploration–exploitation balance. Also, the Newton method is applied to solve the linear quadratic optimal problem. The advantage of applying the method turns out to be computationally fast and has high robustness. For reducing the computational burden, a structured exploitation strategy is proposed. However, the method requires additional computational time while taking high curvatures. Also, for convergence in the 1 optimal solution, more iterations are required.

These UAV path-planning algorithms have their own strengths and weaknesses depending upon the environmental considerations and the computation capacity. Graph search algorithms like Dijkstra and A* provide optimal or nearly optimal paths but need knowledge of the environment beforehand and require map generation as well. Sampling methods, such as Probabilistic Roadmap Planning (PRM) and Rapidly-exploring Random Tree (RRT), on the other hand, represent the efficient approaches for high-dimensional spaces through sampling the configuration space in a random manner. In general, metaheuristics optimization techniques, such as Ant Colony Optimization (ACO) [35] and Particle Swarm Optimization (PSO), among others, are popular tools for global path optimization. However, they tend to be computationally demanding and thus may not be ideal for use in strict real-time applications. The reactive planning techniques include the Dynamic Window Approach (DWA) [36] and Artificial Potential Fields (APF) [37]. These approaches directly process sensor inputs and allow for real-time obstacle avoidance in real time without the need for a global map. However, the reactive planning algorithms are plagued by the problem of local minima and sub-optimal paths among others. Learning-based and soft computing techniques such as fuzzy logic systems, neural networks, and reinforcement learning enable adaptive decision-making through learning from sensory input and interaction with the environment. Such techniques work well in uncertain and changing environments

Table 1 Important path planning algorithms found in the literature

Year	Author(s)	Algorithm Method	Offline / Online	Contribution	Environment
2023	Zu et al. [23]	Improved HPO-based path planning	Offline	Improves UAV path planning performance using an enhanced optimization approach	Multi-task / Complex
2023	Li et al. [24]	UAV path planning for target coverage	Online	Addresses path planning for dynamic target coverage tasks	Dynamic
2021	Chen et al. [25]	Clustering-based path planning	Offline	Proposes clustering-based approach for UAV path planning	Multi-UAV / Complex
2022	Zhang et al. [26]	Dynamic mission planning algorithm	Offline / Online	Develops mission planning strategy for UAV formation	Dynamic
2023	Hu et al. [27]	Distributed multi-UAV coverage method	Online	Presents cooperative path planning for multiple UAVs	Dynamic
2022	Chang et al. [28]	Greedy-based path planning with skeleton extraction	Online	Improves UAV trajectory planning using structured path representation	Structured / Grid
2024	Zhang et al. [29]	RFA-Star (Refined A*)	Offline / Online	Enhances A* algorithm for improved path planning efficiency	High-density Static
2026	Miao et al. [30]	Fuzzy neural fluid dynamical system	Online	Proposes real-time path planning approach using fuzzy-neural modeling	3D / Dynamic
2026	Zhao et al. [31]	A* and Artificial Potential Field (hybrid)	Offline / Online	Combines global and local planning strategies for UAV navigation	Static / Partially Known
2026	Hu et al. [32]	Multi-strategy PSO-based method	Offline	Introduces improved PSO-based approach for UAV path planning	Complex / Cluttered
2026	Gao et al. [33]	Bio-inspired optimization-based path planning	Offline / Online	Proposes optimization-based method for multi-UAV path planning	Complex 3D
2026	Liu et al. [34]	PSO-ABC-based path planning	Offline	Develops hybrid meta-heuristic approach for UAV path planning	Urban / Static

but tend to require large training sets and computational power, making them unsuitable for use in UAVs. The choice of suitable path planning algorithms is influenced by various parameters like awareness of environment (known/unknown), computation capability, real-time considerations, etc. The table below gives a summary of some important path planning algorithms in literature in the years between 2021 and 2026. There seems to be a definite shift towards hybrid path planning algorithms.

Some recent studies (2024–2026) have highlighted hybrid approaches for the navigation of UAVs in challenging conditions. Some examples include the introduction by Zhang et al. [31] of RFA-Star which is an improved A* algorithm with adaptive heuristics for high density of obstacles environment. Another example would be that of Miao et al. [32] who introduced a fuzzy-neural fluid dynamical approach for real time 3D path planning. Then there was the combination of A* with Artificial Potential Fields by Zhao et al. [33] and that of Hu et al. [34]. A fusion-based optimizer for collaborative multi-UAV missions in complex 3D terrains has been suggested by Gao et al. [35]. On the other hand, Liu et al. [36] have considered GIS topology in the PSO-ABC framework to provide practical solutions for urban path planning problems. As a classical reactive baseline, DWA is known to test the feasibility of velocity commands based on kinematics and safety constraints in real-time manner [36]. In contrast, APF approach provides motion commands using the forces of attraction and repulsion [37]. Although efficient, however, these techniques have shown some limitations when applied in very cluttered environments, thereby prompting the need for the creation of more robust adaptive models. In addition, comparisons made between this technique and reactive techniques such as the Dynamic Window Approach (DWA) and the Artificial Potential Field (APF) revealed that the trajectory created by ABCD_LoR is similar to that of DWA, but smoother compared to that of APF. The APF technique gets stuck at local minima in cluttered environments, leading to more detours.

From the comparison table, it can be seen that the current techniques have a trade-off in terms of optimality, efficiency, and adaptiveness. Classical graph-based algorithms like A* give an optimal solution; however, they require the knowledge of the environment beforehand and are not very suitable for real-time applications in dynamic situations. Reactive algorithms like APF and DWA allow quick local decisions but face problems like local minima. Though the fuzzy logic approach is efficient in handling uncertainty, it has a complicated rule-making process and is not scalable. The learning based method gives adaptability and high efficiency under complicated conditions; however, it requires large training data sets and has higher computation costs.

These restrictions prompt the creation of a robust yet simple, statistical, and reactive learning-based approach that ensures trade-offs between real-time performance and decision-making transparency. Reviewing the current literature, several research opportunities arise. The first one involves the absence of common frameworks for integrating and comparing various path planning approaches in different environments. Second, a significant majority of the current approaches are mostly analyzed in a static or semi-static environment and do not perform well when it comes to unpredictable and dynamic conditions. Third, although techniques like soft computing and learning can enhance adaptability, they require a large amount of computational cost, thus

Table 2 Limitations of Existing Methods

Method	Strength	Limitation	Relevance to this work
A*	Optimal and complete path planning with a guaranteed shortest path under an admissible heuristic	Requires prior map knowledge and becomes computationally expensive in large or dynamic environments	Limited suitability for real-time reactive UAV navigation
APF	Computationally efficient and suitable for real-time local obstacle avoidance	Susceptible to local minima and oscillations in cluttered environments	Requires enhancement for robust navigation
DWA	Provides real-time feasibility with dynamic constraints and kinematic awareness	Produces conservative and potentially suboptimal paths with limited global awareness	Serves as a strong reactive baseline
Fuzzy Logic Controllers (FLC)	Effective in handling uncertainty and noisy sensor inputs	Rule design is complex and problem-specific, with limited scalability	Motivates simpler adaptive models
Learning-based methods	Adaptive and capable of learning complex environmental dynamics	Require substantial training data and computational resources, with limited interpretability	Motivates a lightweight and interpretable learning framework

making them inapplicable in case of UAVs with limited resources. Fourth, for the case of graph-based approaches, the problem lies in modeling of the environment.

3 System Model

The ABCD LoR model, which is proposed in this work, is developed to overcome the important drawbacks of the ASGD_LoR algorithm, namely, its susceptibility to noise in update calculations and optimization instability. Indeed, in stochastic gradient algorithms, parameter updates depend on samples very much; therefore, they may vary greatly because of the estimation of coefficients based on one single sample. To deal with such a problem, the ABCD LoR model uses an approach called adaptive block coordinate descent (ABCD). This leads to more robust convergence behavior and makes the algorithm less sensitive to outliers. In contrast to stochastic gradient descent, which performs incremental learning, the block coordinate descent method updates blocks of parameters in an effort to minimize the objective function in relation to one block, while the other blocks remain unchanged. This process is further facilitated by the adaptively chosen update strategy, which ensures that the blocks that help the negative log-likelihood decrease are updated first. In terms of the error variance assumption, logistic regression does not rely on the traditional homoscedasticity

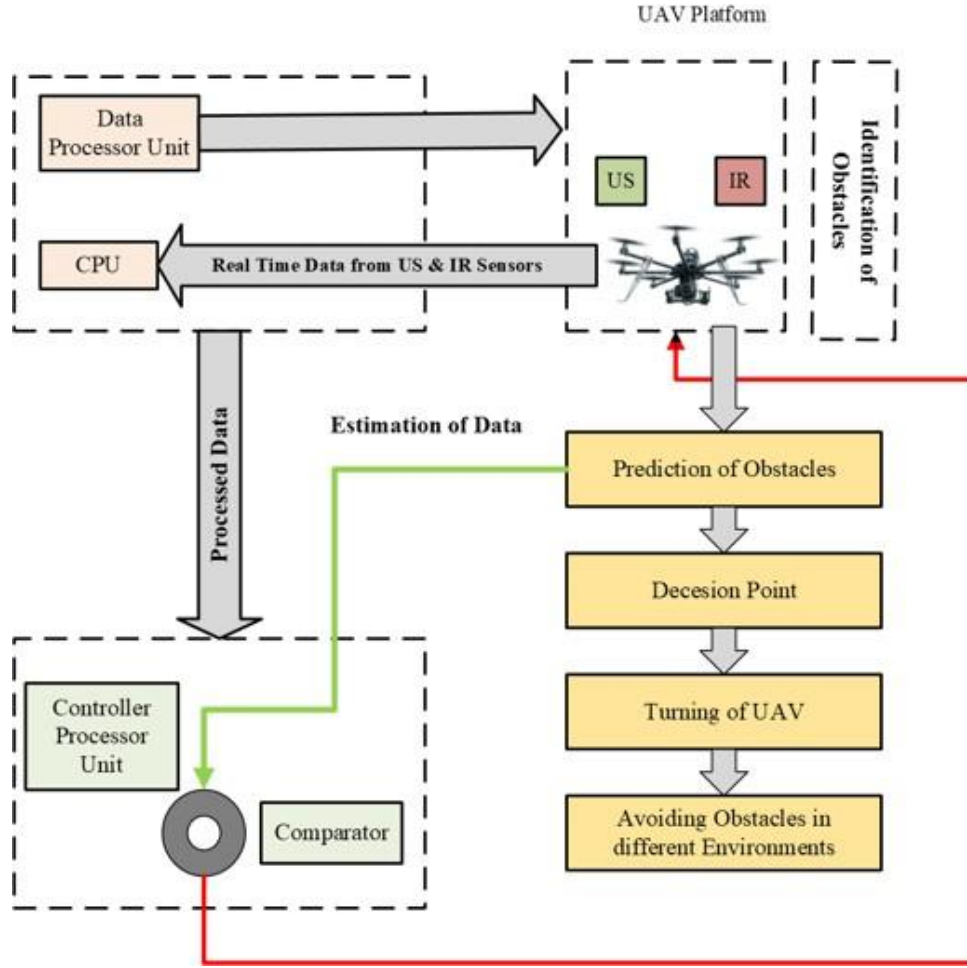


Fig. 1 System architecture of the UAV obstacle avoidance platform with IR and ultrasonic sensors integrated into the ABCD-LoR module.

assumption that is found in the linear model. Rather, it is natural for logistic regression to model the variance of the response as a function of the probability of success using the Bernoulli (or multinomial) distribution. Therefore, the ABCD-LoR model gains from this probabilistic formulation such that class probabilities can be estimated stably without having to make assumptions about the constant error variances. Generally, the use of the probabilistic logistic regression framework along with adaptive block coordinate optimization results in better numerical stability and faster convergence than the baseline ASGD-LoR model. The sequential decision-making approach for obstacle avoidance is shown in Fig. 1.

3.1 System Architecture and Decision Model

In the present UAV system, there are two IR sensors and one US sensor. They detect the proximity of the obstacles in the surroundings. All the sensors are always operational and provide distance information in real time within their individual detection ranges. The data obtained from the sensors are sent to the processing unit for further preprocessing and feature extraction. These feature vectors are fed into the ABCD LoR model which acts as a reactive decision-making module. In order to make the next decision based on the sensor readings available at that instant, this model makes a prediction about the navigation maneuver needed in order to navigate away from the obstacle. The output is sent to the control unit where it computes the motor control signal to correct the UAV direction. This can be any of the three maneuvers: No Turn, Left Turn, or Right Turn.

The probability of selecting a class k is modeled using the softmax function based on the input feature vector. The UAV performs obstacle avoidance by selecting an appropriate action when the measured distance falls below a predefined threshold.

The class probabilities are defined as:

$$P(y = 0 | \mathbf{x}) = \frac{\exp(\mathbf{w}_0^T \mathbf{x})}{\exp(\mathbf{w}_0^T \mathbf{x}) + \exp(\mathbf{w}_1^T \mathbf{x}) + \exp(\mathbf{w}_2^T \mathbf{x})} \quad (1)$$

$$P(y = 1 | \mathbf{x}) = \frac{\exp(\mathbf{w}_1^T \mathbf{x})}{\exp(\mathbf{w}_0^T \mathbf{x}) + \exp(\mathbf{w}_1^T \mathbf{x}) + \exp(\mathbf{w}_2^T \mathbf{x})} \quad (2)$$

$$P(y = 2 | \mathbf{x}) = \frac{\exp(\mathbf{w}_2^T \mathbf{x})}{\exp(\mathbf{w}_0^T \mathbf{x}) + \exp(\mathbf{w}_1^T \mathbf{x}) + \exp(\mathbf{w}_2^T \mathbf{x})} \quad (3)$$

where D denotes the distance to the nearest static obstacle, and v_l and v_r represent the normalized control inputs or motion states of the UAV. The input feature vector is defined as: $\mathbf{x} = [D, v_l, v_r, 1]$

The model estimates the probabilities for three navigation actions: **No Turn (0)**, **Left Turn (1)**, and **Right Turn (2)**, based on the current sensor observations. Fig. 2 illustrates the overall collision avoidance framework. Equations (1) - (3) provide the probabilistic formulation for selecting the appropriate navigation action at each decision step.

Similarly, Equations (4) - (6) represent the same softmax-based probability formulation for the three navigation actions, expressed for completeness of the decision model:

$$P(y = 0 | \mathbf{x}) = \frac{\exp(\mathbf{w}_0^T \mathbf{x})}{\exp(\mathbf{w}_0^T \mathbf{x}) + \exp(\mathbf{w}_1^T \mathbf{x}) + \exp(\mathbf{w}_2^T \mathbf{x})} \quad (4)$$

$$P(y = 1 | \mathbf{x}) = \frac{\exp(\mathbf{w}_1^T \mathbf{x})}{\exp(\mathbf{w}_0^T \mathbf{x}) + \exp(\mathbf{w}_1^T \mathbf{x}) + \exp(\mathbf{w}_2^T \mathbf{x})} \quad (5)$$

$$P(y = 2 | \mathbf{x}) = \frac{\exp(\mathbf{w}_2^T \mathbf{x})}{\exp(\mathbf{w}_0^T \mathbf{x}) + \exp(\mathbf{w}_1^T \mathbf{x}) + \exp(\mathbf{w}_2^T \mathbf{x})} \quad (6)$$

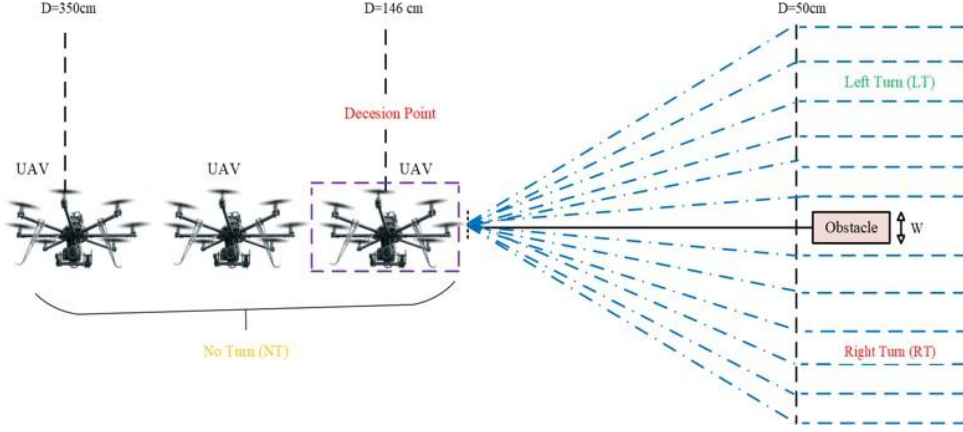


Fig. 2 Reactive collision avoidance framework using sequential three-class logistic regression decisions based on local sensor inputs.

These equations compute the probabilities of selecting one of the three navigation actions: **No Turn (0)**, **Left Turn (1)**, and **Right Turn (2)**, based on the input feature vector.

ABCD LoR model works under various movement conditions and determines the most appropriate navigational action for each time step. The probabilities generated are directly utilized in selecting the action with the highest probability and not converted into binary.

In order to demonstrate how the model behaves, different values of the feature vector x can be put into Equations (1) – (3) and determine the class probabilities. Different parameter values, such as obstacle distance, can be used to investigate the sensitivity of the model under various environmental settings.

Optimization of the model parameters is done through an iterative procedure called the Adaptive Block Coordinate Descent (ABCD) algorithm. In every iteration, a subset of the model's parameters is updated one after another so as to reduce the negative log-likelihood objective function. This process is done until convergence is attained; that can either be achieved through a certain number of iterations (e.g., 100) or a certain log-likelihood value (10^{-5}). The overall optimization workflow is illustrated in Fig. 3.

In relation to the minimization of the negative log-likelihood function of the ABCD LoR, the basic optimization algorithm updates the coordinate blocks B_1 and B_2 through the iterative process. The termination of the algorithm is performed when the maximum number of 100 iterations is reached for the given data set of 292. The model employs a logistic regression model as the base machine learning model for learning the probability distribution over the discrete navigation actions. In contrast to binary logistic regression, the model employs multiclass classification, where the output class belongs to one of the three classes: "No Turn," "Left Turn," or "Right Turn."

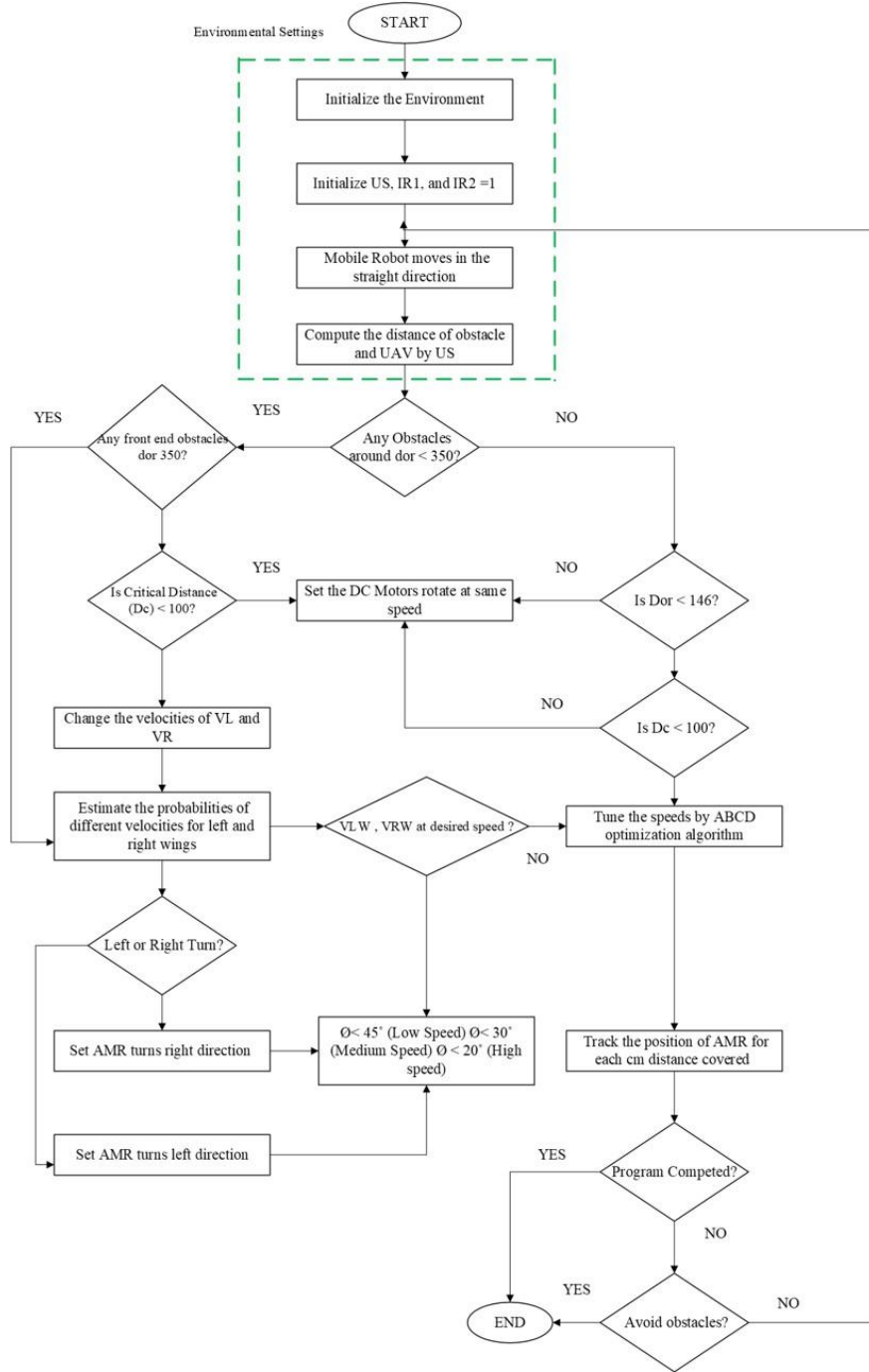


Fig. 3 The proposed ABCD LoR algorithm's flow Diagram

Algorithm 1 Pseudocode for ABCD LoR optimization algorithm

Require: Training dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$; feature vector $\mathbf{x} = [D, v_l, v_r, 1]$; number of classes $K = 3$; maximum iterations T ; convergence threshold ϵ

Ensure: Optimized weight vectors $\mathbf{W} = \{\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2\}$

```
1: Initialize weight vectors  $\mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2$  randomly
2: Partition parameters into coordinate blocks:
3:    $B_1$ : parameters associated with feature  $D$ 
4:    $B_2$ : parameters associated with features  $v_l$  and  $v_r$ 
5: for  $t = 1$  to  $T$  do
6:   Compute predicted probabilities for all samples using softmax:
7:     
$$P(y = k | \mathbf{x}_i) = \frac{\exp(\mathbf{w}_k^T \mathbf{x}_i)}{\sum_{j=0}^{K-1} \exp(\mathbf{w}_j^T \mathbf{x}_i)}$$

8:   Compute negative log-likelihood loss:
9:     
$$L = - \sum_{i=1}^N \sum_{k=0}^{K-1} y_{ik} \log P(y = k | \mathbf{x}_i)$$

10:   Select block adaptively ( $B_1$  or  $B_2$ ) based on the largest gradient magnitude
11:   Update selected block parameters:
12:      $\mathbf{w}_{\text{block}} \leftarrow \mathbf{w}_{\text{block}} - \eta \nabla L(\mathbf{w}_{\text{block}})$ 
13:   Check convergence:
14:   if  $|L^{(t)} - L^{(t-1)}| < \epsilon$  then
15:     break
16:   end if
17: end for
18: return Optimized weights  $\mathbf{W}$ 
```

The ABCD optimization technique updates the subgroups of the model parameters iteratively, which increases stability in the process of convergence and decreases sensitivity to the noise in the gradients in comparison with the stochastic gradient optimization techniques. Model parameters are optimized to maximize classification accuracy, measured by the Receiver Operating Characteristic (ROC) curve under various operation regimes. In order to evaluate the effectiveness of the classification, the suggested algorithm is compared with machine learning classifiers such as Bernoulli Naïve Bayes, Gradient Boosting, and K-Nearest Neighbours. Moreover, the performance in obstacle avoidance is tested in three cluttered environments in comparison with classical path planning algorithms such as ASGD, LoR, VFH, FLC, and A*.

3.2 Dataset Generation and Training Procedure

The training dataset consists of 292 labeled samples generated using a custom 2D grid-based simulation environment. Each sample includes four features:

- (i) distance to the nearest frontal static obstacle measured using an ultrasonic sensor (range: 4 - 400 cm, operating frequency: 40 kHz, beam angle $\approx 15^\circ$),
- (ii) normalized left control input,

- (iii) normalized right control input, and
- (iv) a bias term.

Labels are assigned using predefined deterministic rules based on domain-informed heuristics derived from sensor distance thresholds:

- **No Turn (class 0):** if $D > 120$ cm
- **Left Turn (class 1):** if $D \leq 120$ cm and the left direction indicates greater free space
- **Right Turn (class 2):** otherwise

The dataset has been divided into training (80%, 234 samples) and validation (20%, 58 samples). For the reason of having a smaller-sized dataset, k-fold cross-validation has not been used. The result is averaged based on 10 independent executions with various random initialization of the model parameters. Optimization algorithm ABCD runs up to 100 iterations or until it converges when the difference between the previous negative log-likelihood and current is less than 10^{-5} . There is no regularization in the main experiments except for the fact that from preliminary results there is some gain from using small λ L2 (0.001–0.01). The trained model selects one of the three possible actions (no turn, left turn, and right turn) at fixed time intervals (about 20 to 50 Hz depending on the operational environment). The execution of each action leads to an orientation change of the UAV (left turns by 15° or right turns by 15° , and no turns by 0°), whereas its forward velocity remains unchanged. ABCD_LoR performs multi-class classification for selecting control actions that can be classified into three categories (no turn, left turn, and right turn), using the readings from local sensors; it does not optimize for the path length but ensures safety by making sequential decisions. It should be stated that the suggested approach has a reactive nature and cannot provide global optimality. In comparison with the global path planning approaches, the produced paths will be longer and may oscillate in symmetric environments. Nevertheless, the main purpose of the ABCD_LoR model is to make real-time decisions.

3.3 Adaptive Selection Rule in Block Coordinate Descent

Typical strategies for coordinate descent algorithms follow a predefined or randomized order [38]. However, our approach employs an adaptive selection process based on the block coordinate which will result in maximal decrease of the objective function at each iteration. For each of these blocks, the first order estimation of the decrease of the negative log-likelihood is calculated. Based on the squared norm of the gradient normalized by the block-wise Lipschitz constant estimate, we select the coordinate block which results in the greater estimated decrease. To avoid any stagnation in the procedure, a minimal probability of the selection is imposed. The adaptive selection scheme ensures faster convergence, especially when some attributes (such as obstacle distance) have higher weight on the decision boundary formation process. Unlike gradient descent methods, the suggested technique ensures lower gradient variation and stability. In contrast to Newton or quasi-Newton optimization schemes, the algorithm

is not dependent on Hessian matrix calculation, which makes the technique have linear complexity relative to the number of features. The empirical evaluation carried out in Section 4 shows that the proposed ABCD LoR technique reduces decision time by 20–35% compared to the gradient descent-based approach, while having competitive prediction accuracy.

4 Results and Discussion

The designed ABCD LoR model was tested on a computer with Windows 11 operating system, having an Intel® Core i7 (9th Generation) processor operating at a frequency of 2.60 GHz with 16 GB RAM. Training was performed on 80% of the collected sensor data, with the other 20% reserved for testing purposes. Testing was done by considering three different speed values, namely low (2.77 cm/s), medium (13.8 cm/s), and high (19.4 cm/s). Figure 5 shows the average path lengths achieved after ten experiments, each with standard deviations ± 1 . According to the results, the ABCD LoR algorithm outperforms baseline approaches with a shorter trajectory length in around 25% to 35%. The experiment is conducted in two scenarios at each speed with a variety of environments that include four stationary obstacles with different sizes and shapes. This experiment setup is shown in Fig. 4.

For Case-1, there are obstacles placed in a sequence, such as narrow obstacles and irregular shapes, which need careful decisions at that point. In Case-2, the obstacles have an equal distance apart from each other with one 3D obstacle in between, checking how the algorithm is adaptable to such environments. In Case-3, there are scattered obstacles placed diagonally and frontally. All the algorithms (A*, VFH, FLC, ASGDRL, ABCD LoR, DWA, and APF) were analyzed for ten iterations.

Note: Results averaged over 10 independent simulation runs per condition. DWA and APF values are preliminary and based on standard implementations adapted to the same sensor inputs and dense static layouts.

Performance of all the algorithms in terms of path length (cm) and computational time (ms) is shown in Table 3. In most cases, ABCD LoR provides paths that are shorter than those provided by classic reactive planners like VFH, FLC, DWA, and APF while requiring less computation effort. In dense environments, where the adaptability of the algorithm is vital, the result is more significant. A* provides optimal solutions but is costly in terms of computation and not a reactive planner. However, in this research work, A* is performed on grid maps. A* can find the best route only if it is presented with an accurate map, but cannot be used in a reactive situation where prior environment knowledge is lacking. Conversely, ABCD LoR is based only on local sensor inputs and does not need prior environment knowledge. ABCD LoR is characterized by significantly lesser computation time on a per-decision basis due to its low-dimensional algorithm, resulting in computation times 20–35% lower than A* and VFH under the same simulation conditions. Compared with A* and VFH, the proposed algorithm results in about 20–35% decrease in computation time because of its reduced logistic model. The performance of several UAV path planning algorithms is compared for three speed classes (low, moderate, and high) and two dense environments. The algorithms compared are A*, VFH, FLC, ASGDRL, and the proposed

Table 3 Performance comparison of path planning algorithms for UAVs under varying speed and environmental conditions

S. No.	Speed Condition	UAV Speed (cm/s)	Environment	Algorithm	Path Length (cm)	Computational Time (ms)
1	Low	2.77	Case-1	A*	886.78	3.92
				VFH	731.11	3.23
				FLC	694.31	3.06
				ASGDLR	672.87	2.97
				ABCD _LoR	579.54	2.56
				DWA	615.20	2.85
				APF	695.50	2.92
2	Low	2.77	Case-2	A*	946.82	4.18
				VFH	888.76	3.92
				FLC	764.52	3.37
				ASGDLR	692.73	3.06
				ABCD _LoR	674.48	2.98
				DWA	715.30	3.20
				APF	810.10	3.35
3	Medium	13.8	Case-1	A*	542.67	1.05
				VFH	874.45	0.93
				FLC	776.72	0.82
				ASGDLR	682.28	0.77
				ABCD _LoR	638.85	0.68
				DWA	680.40	0.75
				APF	765.20	0.80
4	Medium	13.8	Case-2	A*	565.47	1.12
				VFH	827.53	0.99
				FLC	756.62	0.91
				ASGDLR	710.14	0.85
				ABCD _LoR	672.27	0.81
				DWA	710.50	0.88
				APF	805.80	0.95
5	High	19.4	Case-1	A*	910.21	0.78
				VFH	871.96	0.74
				FLC	712.78	0.61
				ASGDLR	665.63	0.57
				ABCD _LoR	630.16	0.54
				DWA	670.80	0.60
				APF	755.40	0.65
6	High	19.4	Case-2	A*	951.34	0.81
				VFH	912.24	0.78
				FLC	844.29	0.72
				ASGDLR	770.32	0.66
				ABCD _LoR	691.21	0.59
				DWA	735.60	0.64
				APF	830.90	0.70

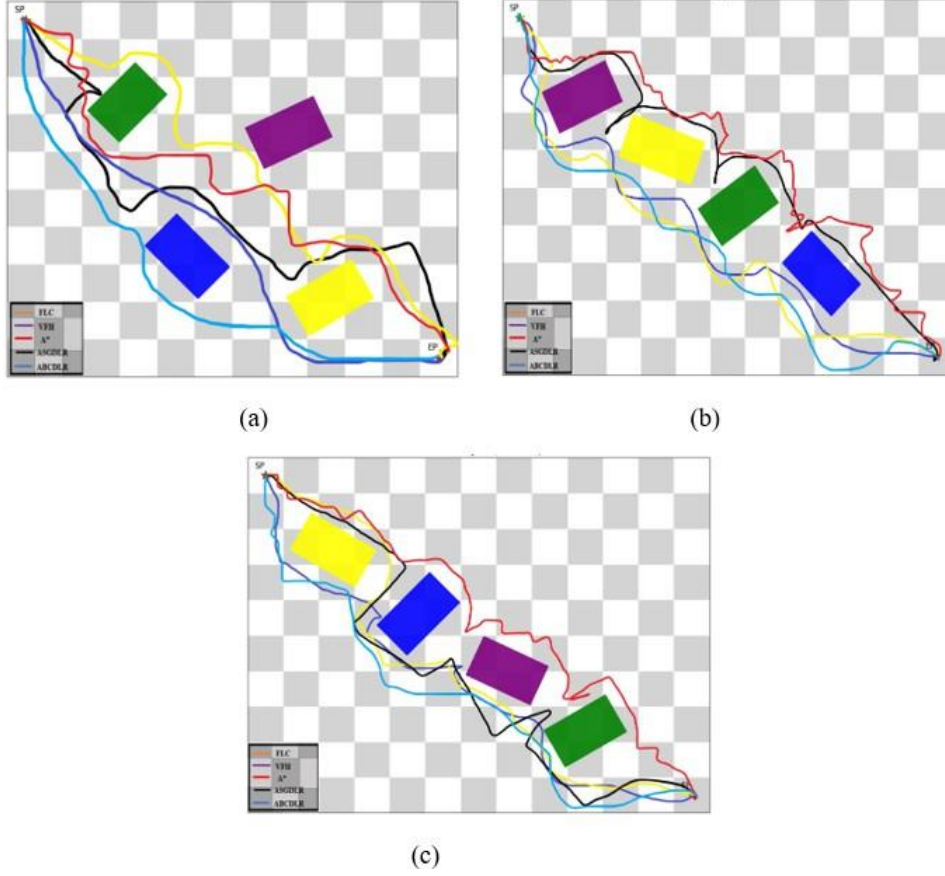


Fig. 4 Reference paths in three dense static environments: (a) Case-1, (b) Case-2, (c) Case-3. SP = start point, EP = end point.

ABCD LoR algorithm. A* ensures shortest paths that are optimal provided a complete map is available but consumes more computational resources and does not react to changing environments. ABC LoR, however, runs on sensor-based decisions only, thus making real-time decisions without relying on any map data. It was found that the model outperforms the reactive baseline methods for all situations through reduced path length and computational cost. These gains can be credited to the lightweight nature and coordinate descent optimization of the model.

The present training data set consists of 292 labeled examples that were gathered within three highly populated simulated environments. The described amount of training data, although being adequate for showing the feasibility and for getting interpretable model parameters, may imply the possible danger of overfitting due to the training on the particular obstacle arrangements and sensor readings observed during data gathering. Due to the small amount of the data, the k-fold cross-validation could

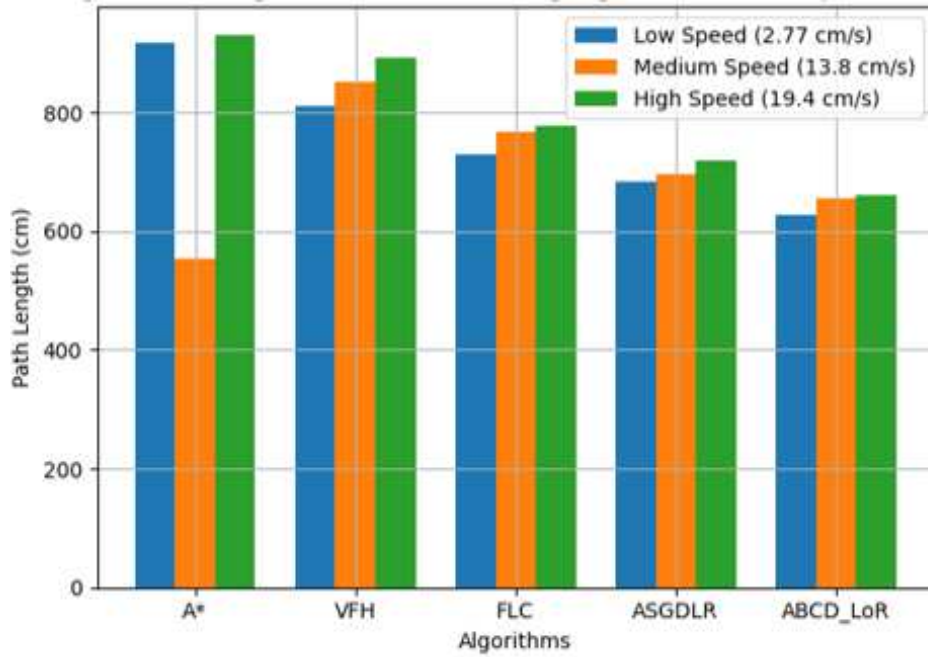


Fig. 5 Average path length (cm) comparison across speeds for ABCD-LoR and baselines (A*, VFH, FLC, ASGDLR). Error bars: ± 1 standard deviation.

not be applied. Rather, the obtained performance results are averaged over ten experiments with various model parameter initialization. Table 4 provides a summary of the benefits and shortcomings of the proposed ABCD LoR algorithm. It is assumed that all tests have been carried out in noise-free conditions for the sensor data. The decision-making process involves a minimal number of calculations, including the application of the softmax function, which guarantees constant complexity per decision.

To test robustness in a practical working environment, future research will involve validation at the hardware level using the quadrotor with respect to the Gazebo/ROS simulated environment. In this testing process, the sensor noise will be considered with Gaussian distribution ($\sigma \sim 5 - 12$ cm) together with measurement dropouts due to reflection effects.

5 Conclusion and Future Work

The following work introduced a robust, fast, and reactive machine-learning based scheme for UAV obstacle avoidance using the adaptive block coordinate descent logistic regression (ABCD LoR) algorithm. The advantage of ABCD LoR lies in overcoming the drawbacks of stochastic gradient algorithms with an optimization technique that involves block coordinate descent algorithm leading to improved stability and resilience to noise. Several experiments conducted on multiple dense static scenarios indicate

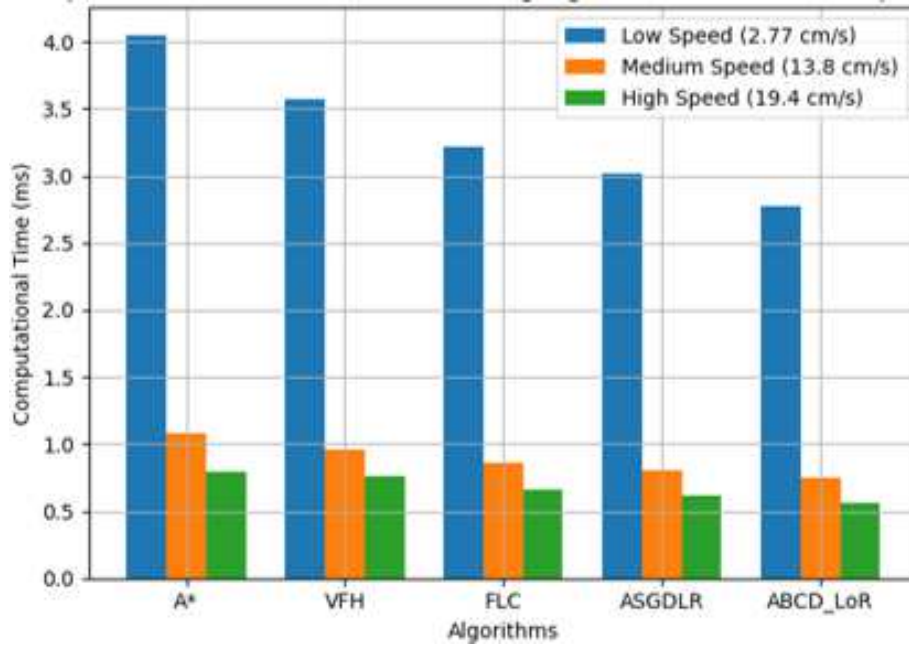


Fig. 6 Computational time (ms) per decision for ABCD-LoR and baselines under low, medium, and high speeds in dense environments

that the proposed model yields shorter path lengths as well as less computational time when compared to traditional techniques like A*, VFH, and FLC. Moreover, comparisons against reactive baselines, such as DWA and APF, show that ABCD_LoR provides smooth paths while still allowing for decision-making in real time. The algorithm works using constant time complexity for each decision and small memory usage, which makes it appropriate for use in UAVs. But the suggested solution is reactive, and therefore it cannot be globally optimal as graph-based planners are. Evaluation in this paper is done using simulation datasets that are relatively small. In addition, the assumption made about sensor readings being free from noise is not true in practice. The future work will be devoted to the extension of the framework presented here in order to solve more practical problems. These are the problems of dealing with dynamic obstacles, adding extra sensors (such as IMU and GPS), and working with UAV hardware. More research on robustness in case of sensor errors, scaling up the size of the data set, and adaptation to fast movement will be done as well. Moreover, research on hybrid approaches that integrate the reactive system with global planners can be done. This will help in optimizing the path while remaining reactive at the same time. The training data set used is made up of 292 data points that have been acquired in simulations with small-sized environments. Thus, there is a danger of overfitting in this case.

Table 4 Advantages and limitations of compared reactive path planning algorithms

Algorithm	Advantages	Limitations
A*	Guarantees optimal shortest path in fully known static environments	Requires complete map; high computational cost for replanning; not suitable for real-time reactive navigation
VFH	Efficient real-time obstacle avoidance; performs well in dense environments	Produces non-smooth/jagged paths; limited global awareness; sensitive to sensor noise
FLC	Handles uncertainty effectively; generates smooth control actions	Rule design is heuristic and problem-dependent; scalability is limited
ASGDLR	Data-driven approach; low inference cost; adapts to environment patterns	High variance due to stochastic updates; limited to binary decisions
ABCD LoR	Lightweight computation ($O(1)$ per decision); interpretable; stable convergence; improved smoothness and efficiency	Dependent on training data quality; purely reactive; no global optimality guarantee
DWA	Considers robot kinematics; reliable real-time collision avoidance	Conservative behavior in narrow spaces; may generate longer paths
APF	Computationally efficient; simple implementation	Prone to local minima; oscillations in cluttered environments; parameter sensitivity

Declarations

Ethics approval and consent to participate

Not applicable.

Consent to publish

Not applicable.

Availability of data and materials

The dataset supporting the findings of this study is presented in this paper.

Competing interests

The authors declare that they have no competing interests.

Funding

Not applicable.

Authors Contribution

PV, KY, AT, RSP, AC are involved in the formation of the proposed content and writing the original manuscript. TB, PB, DRR are involved in suggestions, revisions, and verification of the technique. All authors reviewed and approved the final manuscript.

References

- [1] Wang, Y., Wang, H., Liu, Y., Wu, J., Lun, Y.: 6-dof uav path planning and tracking control for obstacle avoidance: a deep learning-based integrated approach. *Aerospace Science and Technology* **151**, 109320 (2024)
- [2] Xue, Y., Chen, W.: Multi-agent deep reinforcement learning for uavs navigation in unknown complex environment. *IEEE Transactions on Intelligent Vehicles* **9**(1), 2290–2303 (2023)
- [3] Reda, M., Onsy, A., Haikal, A.Y., Ghanbari, A.: Path planning algorithms in the autonomous driving system: A comprehensive review. *Robotics and Autonomous Systems* **174**, 104630 (2024)
- [4] Tao, F., Ding, Z., Fu, Z., Li, M., Ji, B.: Efficient path planning for autonomous vehicles based on rrt* with variable probability strategy and artificial potential field approach. *Scientific Reports* **14**(1), 24698 (2024)
- [5] Gómez Arnaldo, C., Zamarreño Suárez, M., Pérez Moreno, F., Delgado-Aguilera Jurado, R.: Path planning for unmanned aerial vehicles in complex environments. *Drones* **8**(7), 288 (2024)
- [6] Shao, X., Xia, Y., Mei, Z., Zhang, W.: Model-guided reinforcement learning enclosing for uavs with collision-free and reinforced tracking capability. *Aerospace Science and Technology* **142**, 108609 (2023)
- [7] Niu, Y., Yan, X., Wang, Y., Niu, Y.: Three-dimensional collaborative path planning for multiple ucavs based on improved artificial ecosystem optimizer and reinforcement learning. *Knowledge-Based Systems* **276**, 110782 (2023)
- [8] Wang, C., Cheng, C., Yang, D., Pan, G., Zhang, F.: Path planning in localization uncertaining environment based on dijkstra method. *Frontiers in neurorobotics* **16**, 821991 (2022)
- [9] Zhao, Y., Zhang, J., Zhang, C.: Deep-learning based autonomous-exploration for uav navigation. *Knowledge-Based Systems* **297**, 111925 (2024)

- [10] Al-Kaff, A., Armingol, J.M., La Escalera, A.: A vision-based navigation system for unmanned aerial vehicles (uavs). *Integrated Computer-Aided Engineering* **26**(3), 297–310 (2019)
- [11] Pamarthi, V., Agrawal, R.: Unmanned aerial vehicle path planning with hybrid motion algorithm for obstacle avoidance. *Measurement: Sensors* **36**, 101195 (2024)
- [12] Zhang, M., Wang, H., Wu, J.: On uav source seeking with complex dynamic characteristics and multiple constraints: A cooperative standoff monitoring mode. *Aerospace Science and Technology* **121**, 107315 (2022)
- [13] Jiang, W., Lyu, Y., Li, Y., Guo, Y., Zhang, W.: Uav path planning and collision avoidance in 3d environments based on pomdp and improved grey wolf optimizer. *Aerospace science and technology* **121**, 107314 (2022)
- [14] Yan, K., Ma, B.: Mapless navigation based on 2d lidar in complex unknown environments. *Sensors* **20**(20), 5802 (2020)
- [15] Sze, V., Chen, Y.-H., Yang, T.-J., Emer, J.S.: *Efficient Processing of Deep Neural Networks* vol. 51. Springer, ??? (2020)
- [16] Lane, N.D., Bhattacharya, S., Georgiev, P., Forlivesi, C., Jiao, L., Qendro, L., Kawsar, F.: Deepx: A software accelerator for low-power deep learning inference on mobile devices. In: *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 1–12 (2016). IEEE
- [17] Gul, F., Rahiman, W., Alhady, S.N., Ali, A., Mir, I., Jalil, A.: Meta-heuristic approach for solving multi-objective path planning for autonomous guided robot using pso–gwo optimization algorithm with evolutionary programming: F. gul et al. *Journal of Ambient Intelligence and Humanized Computing* **12**(7), 7873–7890 (2021)
- [18] Chen, J., Du, C., Zhang, Y., Han, P., Wei, W.: A clustering-based coverage path planning method for autonomous heterogeneous uavs. *IEEE Transactions on Intelligent Transportation Systems* **23**(12), 25546–25556 (2021)
- [19] Adil, M., Song, H., Jan, M.A., Khan, M.K., He, X., Farouk, A., Jin, Z.: Uav-assisted iot applications, qos requirements and challenges with future research directions. *ACM Computing Surveys* **56**(10), 1–35 (2024)
- [20] Zhang, Z., Wang, S., Chen, J., Han, Y.: A bionic dynamic path planning algorithm of the micro uav based on the fusion of deep neural network optimization/filtering and hawk-eye vision. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **53**(6), 3728–3740 (2023)

- [21] Peng, C., Huang, X., Wu, Y., Kang, J.: Constrained multi-objective optimization for uav-enabled mobile edge computing: Offloading optimization and path planning. *IEEE Wireless Communications Letters* **11**(4), 861–865 (2022)
- [22] Lin, S., Li, F., Li, X., Jia, K., Zhang, X.: Improved artificial bee colony algorithm based on multi-strategy synthesis for uav path planning. *IEEE Access* **10**, 119269–119282 (2022)
- [23] Zu, L., Wang, Z., Liu, C., Ge, S.S.: Research on uav path planning method based on improved hpo algorithm in multitask environment. *IEEE Sensors Journal* **23**(17), 19881–19893 (2023)
- [24] Li, J., Xiong, Y., She, J.: Uav path planning for target coverage task in dynamic environment. *IEEE Internet of Things Journal* **10**(20), 17734–17745 (2023)
- [25] Chen, J., Zhang, Y., Wu, L., You, T., Ning, X.: An adaptive clustering-based algorithm for automatic path planning of heterogeneous uavs. *IEEE Transactions on Intelligent Transportation Systems* **23**(9), 16842–16853 (2021)
- [26] Zhang, J., Cui, Y., Ren, J.: Dynamic mission planning algorithm for uav formation in battlefield environment. *IEEE Transactions on Aerospace and Electronic Systems* **59**(4), 3750–3765 (2022)
- [27] Hu, W., Yu, Y., Liu, S., She, C., Guo, L., Vucetic, B., Li, Y.: Multi-uav coverage path planning: A distributed online cooperation method. *IEEE Transactions on Vehicular Technology* **72**(9), 11727–11740 (2023)
- [28] Chang, J., Dong, N., Li, D., Ip, W.H., Yung, K.L.: Skeleton extraction and greedy-algorithm-based path planning and its application in uav trajectory tracking. *IEEE Transactions on Aerospace and Electronic Systems* **58**(6), 4953–4964 (2022)
- [29] Zhang, W., Li, J., Yu, W., Ding, P., Wang, J., Zhang, X.: Algorithm for uav path planning in high obstacle density environments: Rfa-star. *Frontiers in Plant Science* **15**, 1391628 (2024)
- [30] Miao, F., Li, H.: Real-time path planning for unmanned aerial vehicles in complex three-dimensional dynamic environments using an enhanced fuzzy neural fluid dynamical system. *Engineering Applications of Artificial Intelligence* **164**, 113274 (2026)
- [31] Zhao, J., Zhang, Y., Ning, L., Xiao, X., Bai, C., Zhang, J., Yang, M.: Adaptive path planning of uav based on a* algorithm and artificial potential field method. *Drones* (2504-446X) **10**(2), 93 (2026)
- [32] Hu, G., Huang, F., Houssein, E.H., Wei, G.: Smooth coverage path planning of unmanned aerial vehicles utilizing a novel multi-strategy integrated particle

- swarm optimization. *Advanced Engineering Informatics* **69**, 104071 (2026)
- [33] Gao, X., Wu, W., Zhang, H., Liu, S., Ding, C., Shi, Y.: Cooperative path planning for multi-uavs in complex 3d environments: A multi-strategy fusion improved crested porcupine optimizer. *Knowledge-Based Systems*, 115464 (2026)
 - [34] Liu, Y., Dong, H., Liu, H., Li, H., Zhang, W.: An improved pso-abc path planning algorithm for uavs based on a construction of urban airspace topology with actual gis data. *Scientific reports* **16**(1), 5048 (2026)
 - [35] Al Baroomi, B., Myo, T., Ahmed, M.R., Al Shibli, A., Marhaban, M.H., Kaiser, M.S.: Ant colony optimization-based path planning for uav navigation in dynamic environments. In: 2023 7th International Conference on Automation, Control and Robots (ICACR), pp. 168–173 (2023). IEEE
 - [36] Fox, D., Burgard, W., Thrun, S.: The dynamic window approach to collision avoidance. *IEEE robotics & automation magazine* **4**(1), 23–33 (1997)
 - [37] Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research* **5**(1), 90–98 (1986)
 - [38] Wright, S.J.: Coordinate descent algorithms. *Mathematical programming* **151**(1), 3–34 (2015)