Trajectory Generation by Sparse Demonstration Learning and Minimum Snap-based Optimization

Taoying Xu[1], Haoping She[1,*], Weiyong Si[2], Chuanjun Li[1]

Abstract—**In this paper, dynamic time warping function is used to establish an optimal control system for four-rotor unmanned aerial vehicle (UAV) to learn how to optimize trajectory planning from sparse demonstration. By continuous Pontryagin Differentiable Programming, UAV learns the objective function based on sparse waypoints demonstration. However, due to the small sample data of sparse demonstration learning, there is a problem of low precision, and Pontryagin's Minimum Principle itself has the limitation of easily falling into the local optimal solution. So, this paper adopts the Minimum Snap trajectory algorithm that meets the dynamic constraints of the agent to generate a planned trajectory, to weighted combination with learning trajectory solved based on continuous Pontryagin Differentiable Programming, and the resulting optimized trajectory has the advantages of small demonstration learning difference loss, reasonable time allocation and reasonable planning, so that UAV can have certain generalization capability and optimize a reasonable trajectory with less energy loss. Finally, the feasibility of the proposed method is verified by the simulation experiment of the four-rotor UAV.**

Keywords—dynamic time warping, sparse demonstration learning, minimum snap, continuous Pontryagin Differentiable Programming, trajectory planning

I. INTRODUCTION

In the field of agent control, it is very significant and difficult to make the agent have the ability of autonomous and stable decision when facing unknown conditions, but demonstration learning is one of the solutions. Through demonstration learning, the agent can fully explore the unknown state space and obtain some generalization ability. At present, learning from demonstration has been widely applied to the trajectory planning of unmanned aerial vehicle (UAV) [1], [2], vehicle automatic driving [3], [4], grasp control of robot arm [5], [6] and other fields related to the motion trajectory planning of agents. Besides, reference [7] investigates the research of immersive teleoperation-based learning from demonstration. Based on the demonstration learning, reference [8] realizes the multi-contact tasks of artificial remote correction agents. In [9], through learning a flexible neural energy function, it improves the accuracy of demonstration learning. These novel demonstration methods of human in the loop, in the future, it can also be further realized in the motion control of UAVs.

A. Pontryagin Differentiable Programming

In the process of demonstration learning, the objective function is usually learned by using inverse reinforcement learning or inverse optimal control. While Pontryagin's Minimum Principle (PMP) can be used to solve the functional extremum problem where the control vector set is a bounded closed set. Reference [10] proposed Pontryagin Differentiable Programming (PDP) end-to-end framework for the first time based on Pontryagin's Minimum Principle (PMP). It can be used to study dynamics, optimal polices and control objective functions. Continuous Pontryagin Differentiable Programming (CPDP) is proposed in [11] to apply PDP to real-world scenarios where agents learn from demonstration. Based on sparse waypoints demonstration, the agent can learn the objective function to minimize the difference loss and get the optimal control trajectory parameters. However, due to the limitations of CPDP algorithm itself, there is still much room for improvement in the optimization effect of generated trajectory.

B. Dynamic Time Warping

The dynamic time warping (DTW) function can use the mapping feature of "one-to-many" or "many-to-one" to better describe the mapping relationship between two time series, and greatly reduce the search and comparison time. Demonstration learning involves the presentation timeline and the actual execution timeline, and since there is usually a time deviation between the two, it is necessary to introduce the DTW function. For example，reference [12] uses the hidden Markov model to encode the demonstration trajectory, and multi-dimensional DTW is used to map the key points of the trajectory in time to generate the generalized trajectory. In [11], DTW function is also introduced to establish an optimal control system based on sparse demonstration learning to improve the control efficiency.

C. Minimum Snap

Generally, path planning algorithms such as A* and RRT* do not consider dynamic constraints, so the trajectory is prone to multiple non-smooth break points. However, the Minimum Snap algorithm can effectively overcome the above problems and limit the shape of the trajectory through equality and inequality constraints, which is usually used for obstacle avoidance. Reference [13] proposes an improved Minimum Snap algorithm to apply UAV to trajectory optimization in plant protection scenarios. Applying Minimum snap algorithm to obstacle avoidance, a collision-free trajectory planning using minimum pinch algorithm is proposed in [14]. In addition, Minimum snap is also widely used in trajectory tracking of quadrotor aircraft [15].

In this paper, by minimizing the difference between sparse demonstration and execution, CPDP can learn the tunable parameters of the objective function and the DTW function. Then, based on the tunable parameters, solving the optimal control system of UAV sparse demonstration learning trajectory planning with DTW function. Aiming at the low accuracy of sparse demonstration learning and the limitations of PMP algorithm itself while comprehensively considering the dual indexes of generalization ability and energy loss of UAV trajectory planning, the trajectory

[1] T. Xu, H. She and C. Li are with the School of Aerospace Engineering, Beijing Institute of Technology.

[2]W. Si is with the School of Computer Science and Electronic Engineering, University of Essex.

*Corresponding author is H. She (Email: shehp@bit.edu.cn)

optimization is achieved by combining the planning trajectory generated by Minimum Snap trajectory planning method with the learning trajectory weighting.

Main contributions of this paper:

- By learning the adjustable parameters of DTW function based on CPDP, it can reasonably allocate the time to reach the starting point, the end point and each waypoint. In this paper, this time allocation is also applied to the generation of planning trajectory based on Minimum Snap algorithm.
- In view of the low accuracy of sparse demonstration learning and the defects that Pontryagin's Minimum Principle is easy to fall into the local optimal solution, the Minimum Snap algorithm is used to generate planning trajectory with less energy loss and optimize the final trajectory by weighted combination with the learning trajectory based on sparse demonstration.

II. ESTABLISHMENT OF MATHEMATICAL MODEL OF OPTIMAL CONTROL SYSTEM

Taking a 6-DOF four-rotor UAV with continuous dynamics as the research object, a mathematical model of optimal control system based on sparse waypoints demonstration learning will be established as follows.

State variable x and control variable u are defined according to the motion equation of the four-rotor UAV in SE (3) space:

$$x = [P, V, Q, \omega]^T \tag{1a}$$

$$u = [T_1, T_2, T_3, T_4]^T \tag{1b}$$

where $P \in \mathbb{R}^3$ is the spatial position, $V \in \mathbb{R}^3$ is the speed, $Q \in \mathbb{R}^4$ is the quaternion attitude of UAV, $\omega \in \mathbb{R}^3$ is the angular velocity; T_1, T_2, T_3, T_4 are the thrust forces acting on the four propellers of the four-rotor UAV.

To maximize the demonstration learning effect, the form J of the objective function based on the minimization of difference loss is set as follows:

$$J(p) = \phi\left[x(t_f), p\right] + \int_0^{t_f} F(x(t), u(t), p)dt \tag{2}$$

where ϕ is the terminal cost indicator, F is the process cost indicator, and p is the coefficient vector for calculating the cost function.

By setting the average speed of the UAV, the expected time stamp τ and the expected time range T can be calculated under the condition that the starting point, the end point, and the position information of the demonstration's waypoints are known. Polynomial DTW function is introduced to explain the mapping relationship between the expected timestamp τ and the actual execution time t.

$$t = \sum_{i=1}^{n} \beta_i \tau^i \tag{3}$$

where n is the highest order of the DTW function, $\beta_i (i = 1, \dots, N)$ is the polynomial coefficient corresponding to each order, and it should be noted that $v_\beta(\tau) = \frac{dt}{d\tau} > 0$.

So, the optimal control system equation based on DTW is as follows:

$$\dot{x}(\tau) = v_\beta(\tau) f\big(x(\tau), u(\tau)\big) \tag{4a}$$

$$J(\theta) = \phi[x(T), p] + \int_0^T v_\beta(\tau) F(x(\tau), u(\tau), p)d\tau \tag{4b}$$

where θ is a tunable parameter vector that integrates the coefficient vector of the cost function and the polynomial DTW function.

$$\theta = [p, \beta]^{\mathrm{T}} \tag{5}$$

Therefore, if θ is known, the optimal control system equation (4) established based on DTW can be further solved to obtain the optimal state trajectory x_θ and optimal control u_θ of the UAV at any time based on demonstration learning.

The following will establish a framework for solving the model, which is equivalent to the solving process of θ, and explain the further optimization process of the trajectory learned from the demonstration.

III. MODEL SOLVING AND TRAJECTORY OPTIMIZATION

A. Learning trajectory based on continuous Pontryagin differentiable programming

The derivation process of solving the learning trajectory based on CPDP algorithm is referred to [11]. Firstly, the optimal solution Sol_θ and the mapping function G are defined as follows:

$$Sol_\theta(\tau) = \{x_\theta(\tau), u_\theta(\tau)\} \tag{6a}$$

$$Sol_\theta = \{Sol_\theta(\tau) \mid 0 \le \tau \le T\} \tag{6b}$$

$$G = g(x, u) \tag{7}$$

If the user has demonstrated a total of N waypoints, then the *i-th* keyframe can be defined as $G^*(\tau_i)$ based on the principle of demonstration keyframe optimality, and the expression of the keyframe set $\mathcal{D}$ is as follows:

$$\mathcal{D} = \{G^*(\tau_i) \mid 0 \le \tau \le T, i = 1,2, \cdots, N\} \tag{8}$$

Firstly, θ_0 needs to be randomly initialized before the first iteration, and θ_k is updated by the projection gradient descent method in the $k+1$ iteration.

$$\theta_{k+1} = \mathrm{Proj}_\Theta\left(\theta_k - \eta_k \frac{dL}{d\theta}\Big|_{\theta_k}\right) \tag{9}$$

where η_k represents the learning rate (step size) and L is the cumulative loss.

Then, the optimal control system equation (4) should be solved by the optimal control solver to obtain the optimal solution Sol_θ. Next, using Sol_θ and the loss l corresponding to each waypoint, the cumulative sum L of all waypoints can be calculated.

$$L(Sol_\theta, \mathcal{D}) = \sum_{i=1}^{N} l\left(g^*(\tau_i), g\,(\tau_i)\right) \tag{10}$$

According to the chain rule (11), it is necessary to find the gradient of the single keyframe loss l for each timestamp τ_i relative to the optimal solution $Sol(\tau_i)$, and $Sol_\theta(\tau_i)$ relative to the gradient of the parameter θ, and then sum repeatedly.

$$\left.\frac{dL}{d\theta}\right|_{\theta_k} = \sum_{i=1}^{N} \left.\frac{\partial l}{\partial Sol_\theta(\tau_i)}\right|_{Sol_{\theta_k}(\tau_i)} \cdot \left.\frac{\partial Sol_\theta(\tau_i)}{\partial \theta}\right|_{\theta_k} \tag{11}$$

To minimize the difference loss, the key and the difficulty lies in the use of CPDP solving $\left.\frac{\partial Sol_\theta(\tau_i)}{\partial \theta}\right|_{\theta_k}$.

Using PMP to solve the optimal control system (4), we need to define the Hamiltonian function:

$$H(\tau) = v_\beta(\tau)F(x(\tau), u(\tau), p) + \lambda(\tau)^T v_\beta(\tau) f\big(x(\tau), u(\tau)\big) \tag{12}$$

The equation of state, equation of costate, governing equation and transversal condition of the system can be obtained as follows:

$$\dot{x}_\theta(\tau) = \frac{\partial H}{\partial \lambda_\theta}\big(x_\theta(\tau), u_\theta(\tau), \lambda_\theta(\tau)\big) \tag{13a}$$

$$-\dot{\lambda}_\theta(\tau) = \frac{\partial H}{\partial x}\big(x_\theta(\tau), u_\theta(\tau), \lambda_\theta(\tau)\big) \tag{13b}$$

$$0 = \frac{\partial H}{\partial u}\big(x_\theta(\tau), u_\theta(\tau), \lambda_\theta(\tau)\big) \tag{13c}$$

$$\lambda_\theta(T) = \frac{\partial h_p}{\partial x}\big(x_\theta(T)\big) \; with \; x_\theta(0) = x(0) \tag{13d}$$

To solve the $\left.\frac{\partial Sol(\tau_i)}{\partial \theta}\right|_{\theta_k}$, it is necessary to take the derivative of the tunable parameter vector θ on both sides of the above (13), and obtain the expression of the Pontriagin Maximum Principle in differential form. $\frac{\partial x_\theta}{\partial \theta}$, $\frac{\partial u_\theta}{\partial \theta}$ and $\frac{\partial \lambda_\theta}{\partial \theta}$ are regarded as new state variables, control variables and costate variables respectively, then the Pontryagin maximum principle of differential form can be transformed into the new system of linear quadratic regulator. Reference [9] has proved by deriving the equivalent Raccati model equation, can further calculation to solve the $\left.\frac{\partial Sol_\theta(\tau_i)}{\partial \theta}\right|_{\theta_k}$.

Therefore, the process of solving the learning trajectory based on CPDP algorithm is essentially a two-layer optimization problem. The outer layer iteratively updates θ by minimizing the difference loss L through CPDP, and the inner layer optimizes θ solved by the outer layer, solves the optimal control system based on DTW function, and generates the optimal solution based on demonstration learning Sol_θ. Then the optimal learning trajectory can be obtained.

B. Planning trajectory based on Minimum Snap algorithm

The idea of Minimum Snap algorithm is to use multi-segment polynomials to fit the motion trajectory in multidimensional space, and dimensions are decoupled from each other, so it is easy to calculate. Besides, according to the differential flat characteristics of the four-rotor UAV [16], its snap index corresponds to angular acceleration, which is related to the motor speed, so Minimizing Snap algorithm can achieve the effect of energy saving.

1) Allocate time

When the user demonstrates $(m-1)$ waypoints, adding the start and end point will generate the m segment trajectory. By learning the adjustable parameters of DTW function based on CPDP, it can reasonably allocate the time to reach each segment, so the overall trajectory $P(t)$ can be expressed as follows:

$$P(t) = \begin{cases} [1, t, t^2 \cdots t^n]p_1, t_0 \le t \le t_1 \\ [1, t, t^2 \cdots t^n]p_2, t_1 \le t \le t_2 \\ \cdots \\ [1, t, t^2 \cdots t^n]p_m, t_{m-1} \le t \le t_m \end{cases} \tag{14}$$

where $p_i = \left[p_{i_0}, p_{i_1} \ldots p_{i_n}\right]^T$ represents the *i-th* segment trajectory's coefficient parameter and n is the highest order of the polynomial, which is set to be the same of each segment in this paper.

2) Determine the highest order

Snap in the Minimum Snap algorithm means the fourth derivative of the position, whose form is as follows:

$$s_i(t) = P_i^{(4)}(t) = \left[0,0,0,0,24, \cdots \frac{n!}{(n-4)!}t^{n-4}\right]p_i \tag{15}$$

If $T(n) = \left[0,0,0,0,24, \cdots \frac{n!}{(n-4)!}t^{n-4}\right]$, then:

$$s_i(t) = T(n) \cdot p_i \tag{16}$$

This algorithm needs to restrict the position, velocity, acceleration, and jerk of the start point and end point, and restrict the position of the middle waypoints, while the remaining information is solved through optimization. So, the relationship between the degree of freedom *K* and the number of constraints f corresponding to the *n*-order trajectory in *m* segment is as follows:

$$K = (n+1) \cdot m \tag{17a}$$

$$f = 4 + (m-1) + 4 \tag{17b}$$

$$K \ge f \tag{17c}$$

It can obtain $n \ge \frac{7}{m}$, then to deal with the extreme case of $m = 1$, take $n = 7$ in this paper.

3) Construct the objective function

To ensure smooth connection between adjacent sub-trajectory and limit the trajectory range, it is also necessary to set corresponding equality constraints and inequality constraints. Therefore, the problem is transformed into an optimal solution of multi-segment trajectory parameters satisfying constraints by constructing corresponding objective functions, as follows:

$$min J(T) \; s.t. A_{eq} \cdot p = b_{eq}, A_{ieq} \cdot p \le b_{ieq} \tag{18}$$

$$\begin{aligned} min \; J(T) &= min \int_0^T ((P^{(4)}(t))^2 dt \\ &= min \sum_{i=1}^{m} \int_{t_{i-1}}^{t_i} \big(s_i(t)\big)^2 dt \\ &= min \sum_{i=1}^{m} {p_i}^T \int_{t_{i-1}}^{t_i} \big(T(n)\big)^T \cdot T(n) dt \cdot p_i \\ &= \sum_{i=1}^{m} {p_i}^T Q_i p_i \end{aligned} \tag{19}$$

where $J(T)$ is the objective function, A_{eq} and b_{eq} represent equality constraint matrices and vectors respectively, A_{ieq} and b_{ieq} represent inequality constraint matrices and vectors respectively. And the matrix Q_i is calculated as follows:

$$Q_i = \int_{t_{i-1}}^{t_i} \big(T(n)\big)^T \cdot T(n) dt \tag{20}$$

So, the problem expressed as (18) can be further transformed into a quadratic programming problem, as:

$$min(\frac{1}{2} p^T Q p + q^T p\ )\ s.t. A_{eq} \cdot p = b_{eq}, A_{ieq} \cdot p \le b_{ieq} \tag{21}$$

According to the convex optimization definition, when Hessian matrix Q is a positive semi-definite matrix, the problem is a convex quadratic programming problem.

4) Set equation constraints

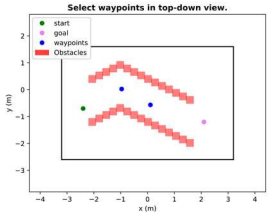
It is usually necessary to set two kinds of equality constraints, namely derivative constraints, and continuity constraints. Among them, derivative constraints at least realize the position, speed, acceleration and jerk constraints on the starting point and end point of the trajectory, and the position constraints on waypoints points. Continuity constraints realize smooth connections between adjacent sub-trajectory, that is, the derivatives of each order at the connection points are the same.

IV. Simulation Experiments

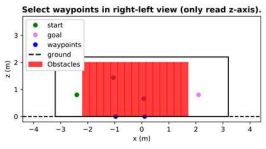
In this experiment, the UAV demonstration learning based on sparse waypoints is realized. Besides, the learning trajectory is weighted with the planned trajectory generated by the Minimum Snap algorithm to further obtain the optimized trajectory.

A. Setting of initial experimental conditions

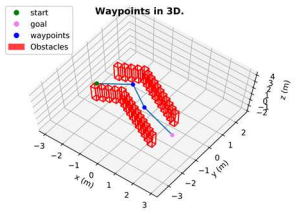
Firstly, several sparse waypoints need to be artificially selected as demonstration data according to the starting point, end point and the position of obstacles in the visual view. In this experiment, two waypoints are selected in total, and the effects of the generated demonstration data are shown in Fig. 1.



(a) waypoints in XOY plane



(b) waypoints in XOZ plane



(c) waypoints in 3D space

Fig. 1. Waypoints are selected artificially from the XOY plane and XOZ plane in turn to obtain the position information of the waypoints in 3D space as the demonstration data, and the demonstration trajectory is generated by linear interpolation.

The initial conditions of this experiment are set as follows in TABLE I. , and all dimensional variables are based on the International System of Units (so does the TABLE II.):

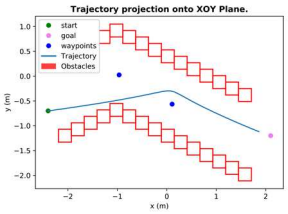
TABLE I. Initial Experimental Condition

Symbol	Physical Meaning	Value
m	Quality of UAV	1.0
l	Distance of each motor from the center of mass	1.0
c	Rotor parameter	0.02
I	Rotational inertia	[1.0,1.0,1.0]
$\bar{v}$	Average flying speed	[1.0,1.0,1.0]
x_0	Initial state variable	[[−2.4, −0.7,0.8], [0,0,0], [1,0,0,0], [0,0,0]]
x_g	Terminal state variable	[[2.1, −1.2,0.8] [0,0,0], [1,0,0,0], [0,0,0]]
$waypoints$	Waypoints position information	[−0.964, 0.025, 1.438], [0.107, −0.564, 0.657]
l_r	Learning rate	0.08
$Itermax$	Maximum number of iterations	80
$Iter$	The actual number of iterations	42

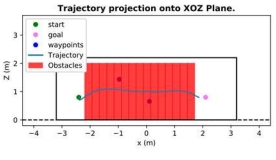
B. Simulation experiment and results

After setting the initial conditions and obtaining the demonstration data, the next step is to learn how to make reasonable trajectory planning from the sparse demonstration based on the CPDP algorithm and generate a learning trajectory.

As explained in III.A section, the essence of this sparse demonstration learning process is a two-layer optimization problem. The outer layer iteratively learns the tunable parameter pair $[p, \beta]^{\mathrm{T}}$ based on CPDP algorithm, and the inner layer uses $[p, \beta]^{\mathrm{T}}$ to solve the optimal control system based on TWD function, updates the optimal UAV state variable and control variable, and combines to obtain a learning trajectory as shown in Fig. 2.



(a)learning trajectory in XOY plane



(b)learning trajectory in XOZ plane

Fig. 2. Based on continuous Pontryagin Differentiable Programming, UAV learns the cost function and dynamic time warping function from the sparse demonstration, and generates a learning trajectory by obtaining the learned optimal state and optimal control finally.

The experimental data output results related to Fig. 2 are recorded in the TABLE II. and the iteration loss is shown in Fig. 3.

TABLE II. OUTPUT RESULT

Symbol	Physical Meaning	Value
$Time\_list$	Time allocation table	$[0.0, 1.73, 3.18, 5.28]$
$loss[max, min]$	Iterative loss interval	[5.313733907871411, 0.6368476895959516]
p	Coefficient vector of the cost function	[3.966453919463763, 1.9248463346416484, −0.043705433571873044, 1.7266226614915658, 0.6576000376522613, 0.8386555889432388, −0.7892306661123814]
β	Coefficient of the DTW function	3.966453919463763

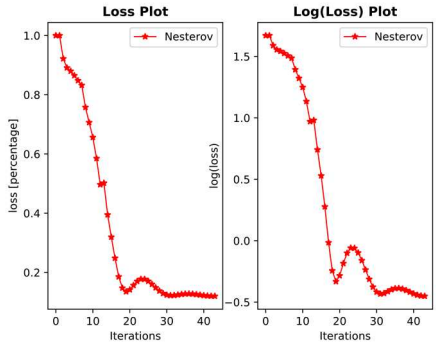
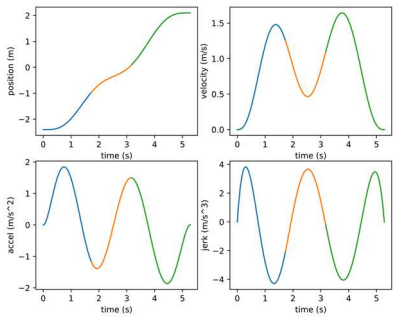


Fig. 3. Updating the tunable parameter pair $[p, \beta]^{\top}$ is achieved by minimizing the difference loss between the sparse demonstration and the actual execution during iterating. The difference iteration loss in the iterative process of learning trajectory as shown in Fig.2 is recorded.

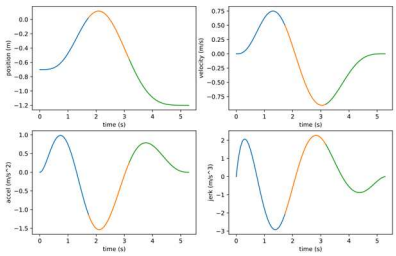
Although it can be seen from the iterative loss in Fig. 3 that after about 30 iterations, the difference loss tends to a lower stable value and almost only changes around 0.1, the learning trajectory is still quite different from the sparse demonstration data. The Fig. 2 shows that the learning trajectory has poor learning effect at the first waypoint, and is very close to the red obstacle area, which has the risk of collision. However, the possible reason is that the limitation of CPDP algorithm itself or the lack of demonstration data will affect the learning effect.

Therefore, to further improve the effect of demonstration learning and avoid collision with obstacles as much as possible, the Minimum Snap algorithm is used to generate a planning trajectory. As mentioned in III.B section, Minimum Snap algorithm can obtain the position, velocity, acceleration and jerk information of the intermediate point at any moment by optimization process with less energy loss (as shown in Fig. 4).

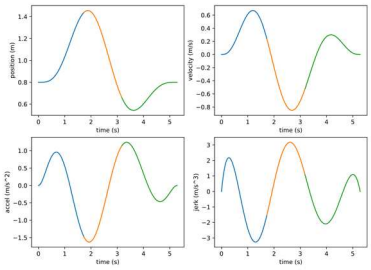
From Fig. 4, it shows the result of simulated UAV flight trajectory in three-dimensional space, where (a), (b) and (c) are respectively the X, Y and Z optimized position, speed, acceleration, and jerk curve effect of Minimum Snap algorithm in one-dimensional space, and Fig. 5 shows the optimized planning trajectory generated in XOY plane and XOZ plane respectively. By comparing Fig. 2 with Fig. 5, it can be concluded that the planning trajectory generated by the Minimum Snap algorithm has a higher security performance than the learning trajectory generated by spare demonstration.



(a) X-axis trajectory

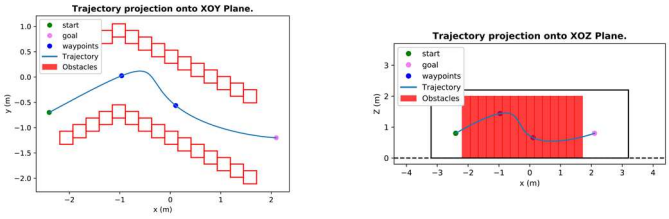


(b) Y-axis trajectory



(c) Z-axis trajectory

Fig. 4. With the starting and ending states of the UAV and the position information of the waypoints as constraints, the position, velocity, acceleration and jerk of the UAV to each intermediate point can be optimized by the Minimum Snap algorithm, and all dimensions of the three-dimensional space are decoupled.

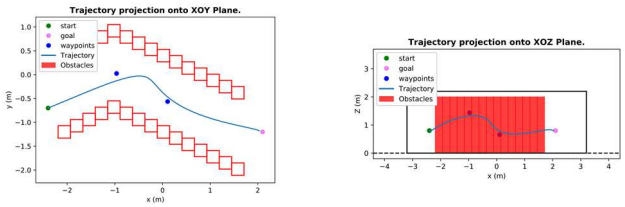


(a)planning trajectory in XOY plane (b)planning trajectory in XOZ plane

Fig. 5. The planning trajectory (in XOY plane and XOZ plane) generated by the Minimum Snap algorithm can fit waypoints well, and can effectively avoid collision with known obstacles.

However, sparse demonstration learning can enable the UAV to have certain generalization ability when facing the unknown state space, and the Minimum Snap algorithm can improve the accuracy of UAV trajectory planning with less capability loss. Therefore, the learning trajectory and planning trajectory are combined by weighting, and the weight coefficient adopted in this experiment is [0.5,0.5]. The resulting optimization trajectory is shown in Fig. 6, and

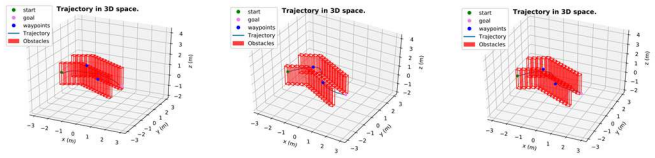
the effect of the final optimization trajectory is superior to that of a single learning trajectory.



(a) final optimization trajectory in XOY plane

(b) final optimization trajectory in XOZ plane

Fig. 6. When the weight coefficient matrix is selected as [0.5,0.5] and under the same and reasonable time allocation, the learning trajectory and the planning trajectory are combined by weighting to generate a final optimization trajectory.



(a)learning trajectory in 3D space

(b)planning trajectory in 3D space

(b)final trajectory in 3D space

Fig. 7. The effects of learning trajectories, planning trajectories and final optimization trajectories generated in this experiment are compared in 3D space.

In short, the trajectory safety after optimization is improved, but the function of realizing a safe flight area and avoiding known obstacles by adding inequality constraints can be considered further.

V. Conclusion

In this paper, CPDP is adopted to realize the double-layer optimal control problem of UAV learning trajectory planning from sparse demonstration. The outer layer iteratively learns the adjustable parameters of objective function and DTW function by gradient descent method, while the inner layer obtains the optimal state and optimal control of UAV at any time by solving the optimal control system. However, through experiments, it is found that the effect of sparse demonstration learning will be affected by the limitations of demonstration data and PDP algorithm itself which will lead to the low learning effect.

Therefore, the time allocation based on DTW function is used in the Minimum Snap algorithm to generate a planned trajectory with small energy loss and passing through each waypoint, and it is weighted with the learning trajectory. Finally, based on experiments, it is verified that the final optimized trajectory can optimize the effect of the learning trajectory and the generalization ability, learning effect and capability loss indexes of UAV in trajectory planning are considered comprehensively meanwhile.

References

[1] C. Zhang, Y. Liu and Z. Zhang, "A Deep Reinforcement Learning Approach for Federated Learning Optimization with UAV Trajectory Planning," 2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Toronto, ON, Canada, 2023, pp. 1-7.

[2] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton and J. Henry, "Joint Cluster Head Selection and Trajectory Planning in UAV-Aided IoT Networks by Reinforcement Learning With Sequential Model," in IEEE Internet of Things Journal, vol. 9, no. 14, pp. 12071-12084, 15 July15, 2022.

[3] M. Kuderer, S. Gulati and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 2015, pp. 2641-2646.

[4] N. Dang, T. Shi, Z. Zhang, W. Jin, M. Leibold and M. Buss, "Identifying Reaction-Aware Driving Styles of Stochastic Model Predictive Controlled Vehicles by Inverse Reinforcement Learning," 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), Bilbao, Spain, 2023, pp. 2887-2892.

[5] M. Kaiser and R. Dillmann, "Building elementary robot skills from human demonstration," Proceedings of IEEE International Conference on Robotics and Automation, Minneapolis, MN, USA, 1996, pp. 2700-2705 vol.3.

[6] O. Fernandez-Ramos, D. Johnson-Yañez and W. Ugarte, "Reproducing arm movements based on Pose Estimation with robot programming by demonstration," 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI), Washington, DC, USA, 2021, pp. 294-298.

[7] Weiyong Si;Ning Wang;Chenguang Yang.A review on manipulation skill acquisition through teleoperation - based learning from demonstration[J].Cognitive Computation and Systems,2021,Vol.3(1): 1-16.

[8] Weiyong Si;Yuan Guan;Ning Wang.Adaptive Compliant Skill Learning for Contact-Rich Manipulation With Human in the Loop[J].IEEE Robotics and Automation Letters,2022,Vol.7(3): 5834-5841.

[9] Zhehao Jin;Weiyong Si;Andong Liu;Wen-An Zhang;Li Yu;Chenguang Yang.Learning a Flexible Neural Energy Function With a Unique Minimum for Globally Stable and Accurate Demonstration Learning[J].IEEE Transactions on Robotics,2023,Vol.39(6): 1-19.

[10] Wanxin Jin;Zhaoran Wang;Zhuoran Yang;Shaoshuai Mou.Pontryagin Differentiable Programming: An End-to-End Learning and Control Framework[J].2020.

[11] W. Jin, T. D. Murphey, D. Kulić, N. Ezer and S. Mou, "Learning From Sparse Demonstration," in IEEE Transactions on Robotics, vol. 39, no. 1, pp. 645-664, Feb. 2023.

[12] A. Vakanski, I. Mantegh, A. Irish and F. Janabi-Sharifi, "Trajectory Learning for Robot Programming by Demonstration Using Hidden Markov Model and Dynamic Time Warping," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 42, no. 4, pp. 1039-1052, Aug. 2012.

[13] Shen, Yue.Optimizing the flight path of plant protection UAV using improved minimum SNAP[J].Nongye Gongcheng Xuebao /Transactions of the Chinese Society of Agricultural Engineering,2023,Vol.39(17): 51-59.

[14] Zhao, Xin;Wang, Ke;Wu, Sixian;Wen, Long;Chen, Zhibo;Dong, Liang;Sun, Mengyao;Wu, Caicong.An obstacle avoidance path planner for an autonomous tractor using the minimum snap algorithm.[J].Computers & Electronics in Agriculture,2023,Vol.207.

[15] Youkyung Hong, Suseong Kim, Yookyung Kim, Jihun Cha.Quadrotor path planning using A* search algorithm and minimum snap trajectory generation[J].ETRI Journal,2021,Vol.43(6): 1013-1023.

[16] Mellinger, Daniel;Kumar, Vijay.Minimum Snap Trajectory Generation and Control for Quadrotors[A].2011 IEEE International Conference on Robotics and Automation[C],2011.