Multiple Shooting Parameterized Differential Dynamic Programming for Waypoint-Trajectory Optimization

Feiran Xia[1], Runqi Chai[1], Chaoyong Li[2], and Shaoming He[1,*]

[1] Beijing Institute of Technology, Beijing 100081, China
[2] Zhejiang University, Hangzhou, 310027, China
*Email: shaoming.he@bit.edu.cn

Abstract. In missions such as reconnaissance, unmanned aerial vehicles (UAVs) are required to rapidly fly through several waypoints and reach the target position. It is therefore necessary to ensure that the trajectory is constrained to pass through the intermediate points and the flight time is optimized. This paper proposes a multiple shooting parameterized differential dynamic programming (MSPDDP) algorithm. The validity of the MSPDDP algorithm is demonstrated through simulations of UAV multiple waypoint reconnaissance. Moreover, the trajectories optimized by MSPDDP are smoother and have a smaller inaccuracy than the MSDDP algorithm, which cannot optimize the time of flight.

Keywords: differential dynamic programming, trajectory optimization, UAV, waypoints

1 Introduction

Unmanned aerial vehicles (UAVs) were first applied in military reconnaissance missions during the Vietnam War, and have since been gradually applied in various fields, such as monitoring and surveying, aerial photography and filming [1], search and rescue [2], logistics and transport, and other civil fields. When performing missions such as reconnaissance, UAVs need to rapidly fly through several designated waypoints and reach the target position.

Optimal control theory can be traced back to the famous maximum velocity descent problem proposed by Johann Bernoulli in the 17th century [3]. Euler first introduced the variational method for solving optimal control problems in 1733 [4], and in the 1950s Richard Bellman pioneered the dynamic programming method, which uses the Hamilton-Jacobi-Bellman (HJB) equation to derive sufficient conditions for optimality [5]. Pontryagin developed the principle of minima (maxima) in 1962, which provides a bang-bang control solution to the constrained optimal control problem [6].

Due to its rapid convergence characteristics, the differential dynamic programming trajectory optimization method based on Bellman's optimality principle has attracted considerable attention and research in recent years [7]. The computation of DDP takes place in a second-order neighborhood of the nominal trajectory, where the control input optimal solution is obtained by continuous forward and backward computation [8, 9]. The main advantage of DDP shows in that it ensures a linear complexity proportional to the time horizon of the prediction and provides the same theoretical second-order convergence speed as Newton's method [10].

In this paper, we propose a new algorithm : multiple shooting parameterized differential dynamic programming (MSPDDP). This proposed algorithm can satisfy the waypoint constraints besides optimize the flight time. In order to address the problem that presetting a fixed flight time may lead to a tortuous flight trajectory, the algorithm converts the flight time between the trajectory points into a time-invariant parameter.

The rest of the paper is organized as follows. Section 2 presents the research work related to the algorithm proposed in this paper. In Section 3, the general form of the trajectory optimization problem is formulated. Then in Section 4, the main calculating processes of MSPDDP algorithm is introduced, a detailed mathematical derivation is given and numerically evaluated in Section 5. Section 6 summarizes the paper and indicates the direction of future work.

2 Related Work

2.1 Differential Dynamic Programming

In recent years, many researchers have developed improvements to the DDP method for different constraints. Tassa et al [11] demonstrated that the use of a control clamping strategy to enforce constraints on the control variables reduces the convergence performance of the algorithm. There are two ways of dealing with the state constraint problem. One is to consider the necessary conditions for optimality and deal with the constraints through the activity set approach [12]. Xie et al [13] proposed the constrained DDP (CDDP) method that can adapt to arbitrary state-control nonlinear inequality constraints.

Another way to deal with the constraint problem is to convert the state equality and inequality constraints into soft constraints and merge them into the objective function through numerical methods such as penalty methods, and to deal with the constraints by using the penalty function method or the multiplier method [14]. Lantoine et al [15] proposed a hybrid differential dynamic programming (HDDP) algorithm, which solves the constrained quadratic programming subproblems using the augmented Lagrangian function, and handles the path constraints using the active set approach. Aoyama et al [16], on the other hand, based on Xie et al [13], utilized the augmented Lagrange multiplier method to compute robustness against initial guesses in the initial stage. Oshin et al [17] proposed a parameterized differential dynamic programming (PDDP) algorithm, which has the ability to optimize the parameter besides control input. However, the algorithm cannot consider the path state constraints, and it cannot handle the time optimization of multiple trajectory segments.

2.2 Waypoint-Trajectory Optimization

In order to solve the problem of multiple designated waypoints in the flight trajectory, a kind of idea is to divide the trajectory into multiple sub-intervals of the trajectory segments, and optimize the trajectory through the idea of multiple shooting method, as Figure 1. Giftthaler et al [18] extended iLQR method to the multiple shooting framework. In order to increase the calculating efficiency and to enhance the robustness of the algorithm to initial guesses when designing orbits of a spacecraft, Etienne et al [19] incorporate the multiple shooting method into the differential dynamic programming framework.

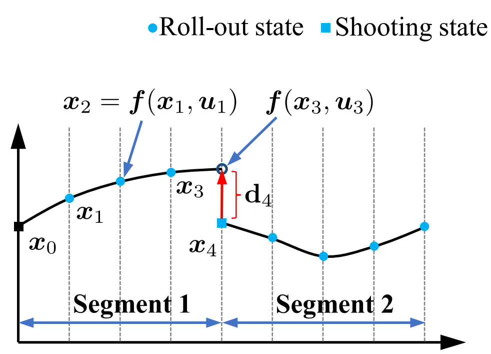

Fig. 1: Multiple shooting method for solving the waypoint-trajectory problem

As for penalty methods, Philipp et al [20] proposes a definition of waypoint proximity that uses a penalty function to push the trajectory closer to a waypoint. Zheng et al [21] proposed a constrained parameterized DDP (C-PDDP) algorithm considering multiple waypoint constraints. The algorithm considers multiple waypoints as one type of path state constraints, and incorporates the intermediate waypoint constraints into the original cost function through penalty function and then ensure that the UAV trajectory passes these intermediate waypoints.

3 Problem Formulation

This section gives a model of free-time waypoint-trajectory optimization problem. This kind of problem usually needs to achieve two objectives, i.e., constraining the trajectory of the UAV to pass through waypoints while enabling time-of-flight optimization of every trajectory segment and keeping cost function as small as possible. Assume that there are W waypoints $\boldsymbol{p}_{w_j}$, in which $j \in [1, \ldots, W]$. Then the indicator to be optimized in free-time waypoint-trajectory optimization problem will be formulated in the form of

$$\min_{\boldsymbol{u},t}\ \Phi\left(\boldsymbol{x}\left(t_f\right), t_f\right)+\sum_{j=1}^{W}\left[\int_{t_{w_{j-1}}}^{t_{w_j}} L_j(\boldsymbol{x}(t), \boldsymbol{u}(t)) \mathrm{dt}\right] \tag{1}$$

where t_f is the terminal time for entire flight and t_{w_j} refers to the terminal time for trajectory segment $\boldsymbol{X}_{w_{j-1} w_j}$.

Under the discrete time frame, the objective function (1) is equally written as

$$\min_{\boldsymbol{u}_k,t}\ \phi\left(\boldsymbol{x}_N, t_N\right)+\sum_{j=1}^{W}\left[\sum_{k=t_{w_{j-1}}}^{t_{w_j}} l_j\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)\right] \tag{2}$$

In a free-time waypoint-trajectory optimization problem, the flight time in every period needs to be calculated to the optimum. The flight time of the UAV between waypoints is recorded as

$$t_{w_j w_{j+1}} \in \mathcal{T} \tag{3}$$

In summary, the UAV free-time waypoint-trajectory optimization problem with simultaneous consideration of control limits can be written as

$$\begin{aligned}
\boldsymbol{u}_k^* = \min_{\boldsymbol{u}_k,t}\ & \phi\left(\boldsymbol{x}_N, t_N\right)+\sum_{j=1}^{W}\left[\sum_{k=t_{w_{j-1}}}^{t_{w_j}} l_j\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right)\right] \\
\text{s.t.} \quad & \boldsymbol{x}_{k+1}=\boldsymbol{f}\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right) \\
& \left\|\boldsymbol{x}_{w_j}-\boldsymbol{p}_{w_j}\right\|_2=0 \\
& \boldsymbol{u}_{\min} \leq \boldsymbol{u}_k \leq \boldsymbol{u}_{\max} \\
& t_{w_j w_{j+1}} \in \mathcal{T}
\end{aligned} \tag{4}$$

4 Multiple-Shooting PDDP for Waypoint-Trajectory Optimization

4.1 Mathematical Derivation of MSPDDP

The original PDDP algorithm is able to optimize the flight time but cannnot constrain the ability of the trajectory to pass through various intermediate waypoints. Improved on the basis of the PDDP algorithm, the MSPDDP algorithm separates the trajectory into multiple trajectory segment subintervals, and conducts trajectory optimization through the idea of the multiple shooting method.

The multiple shooting method transcribes a continuous-time optimal control problem into a discrete-time optimal control problem, i.e.

$$\begin{aligned}
& \min_{\boldsymbol{U};\boldsymbol{\Theta}} J(\boldsymbol{U};\boldsymbol{\Theta})=\sum_{k=0}^{N-1} \ell_k(\boldsymbol{x}_k, \boldsymbol{u}_k;\boldsymbol{\theta})+\phi(\boldsymbol{x}_N;\boldsymbol{\theta}) \\
& \text{s.t.} \quad \underbrace{\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k;\boldsymbol{\theta})-\boldsymbol{x}_{k+1}}_{\mathbf{d}_{k+1}}=0
\end{aligned} \tag{5}$$

in which $0 \leq k \leq N$ refers to the time index. $\boldsymbol{U} = \{\boldsymbol{u}_0, \ldots, \boldsymbol{u}_k, \ldots, \boldsymbol{u}_{N-1}\}$ and $\boldsymbol{\Theta} = [\theta_1, \ldots, \theta_j, \ldots, \theta_W]$, where $\boldsymbol{\theta}$ denotes flight time in this paper. $\ell_k(\boldsymbol{x}_k, \boldsymbol{u}_k; \boldsymbol{\theta})$ is running cost and $\phi(\boldsymbol{x}_N; \boldsymbol{\theta})$ denotes terminal cost. $\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k; \boldsymbol{\theta})$ represents the discrete kinetic equations, and $\mathbf{d}_{k+1}$ denotes the defect function which estimates the dynamic feasibility of $(\boldsymbol{x}_k, \boldsymbol{u}_k; \boldsymbol{\theta}, \boldsymbol{x}_{k+1})$. Equation (5) consider both state and control as decision variables. In this paper, we denote the small perturbations in the neighborhood of nominal trajectory $(\boldsymbol{X}, \boldsymbol{U}; \boldsymbol{\Theta})$ as $(\delta \boldsymbol{X}, \delta \boldsymbol{U}; \delta \boldsymbol{\Theta})$, i.e.

$$\begin{aligned} \delta \boldsymbol{x} &= \hat{\boldsymbol{x}}_k - \boldsymbol{x}_k \\ \delta \boldsymbol{u} &= \hat{\boldsymbol{u}}_k - \boldsymbol{u}_k \\ \delta \boldsymbol{\theta} &= \hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}_k \end{aligned} \tag{6}$$

where $\hat{(\cdot)}$ is the new trajectory. The system dynamics equations in discrete form can be obtained by linearization as

$$\delta \boldsymbol{x}_{k+1} = \delta \boldsymbol{x}_k + (\boldsymbol{f}_{\boldsymbol{x}} \delta \boldsymbol{x} + \boldsymbol{f}_{\boldsymbol{u}} \delta \boldsymbol{u} + \boldsymbol{f}_{\boldsymbol{\theta}} \delta \boldsymbol{\theta}) \Delta t \tag{7}$$

For simplicity of representation, we denote $\Phi = I + \boldsymbol{f}_{\boldsymbol{x}} \Delta t$, $\beta = \boldsymbol{f}_{\boldsymbol{u}} \Delta t$ and $\gamma = \boldsymbol{f}_{\boldsymbol{\theta}} \Delta t$. In practical calculations, the simplified form of Equation (7) is usually used, namely

$$\delta \boldsymbol{x}_{k+1} = \Phi_k \delta \boldsymbol{x}_k + \beta_k \delta \boldsymbol{u}_k + \gamma_k \delta \boldsymbol{\theta} \tag{8}$$

The value function of the MSPDDP considering small perturbations is

$$v(\delta \boldsymbol{x}_k; \delta \boldsymbol{\theta}) = \min [\underbrace{\delta \ell(\delta \boldsymbol{x}_k, \delta \boldsymbol{u}_k; \delta \boldsymbol{\theta}) + v(\delta \boldsymbol{x}_{k+1}; \delta \boldsymbol{\theta})}_{Q(\delta \boldsymbol{x}_k, \delta \boldsymbol{u}_k; \delta \boldsymbol{\theta})}] \tag{9}$$

where $v(\delta \boldsymbol{x}_k; \delta \boldsymbol{\theta})$ refers to the value function, and $\delta \ell(\delta \boldsymbol{x}_k, \delta \boldsymbol{u}_k; \delta \boldsymbol{\theta})$ indicates second-order approximation of the operating cost perturbation in the neighborhood of the nominal trajectory. $(\delta \boldsymbol{X}, \delta \boldsymbol{U})$ is required to satisfy

$$\delta \boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k + \delta \boldsymbol{x}_k, \boldsymbol{u}_k + \delta \boldsymbol{u}_k; \boldsymbol{\theta}_k + \delta \boldsymbol{\theta}_k) - \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k; \boldsymbol{\theta}_k) \tag{10}$$

This section reviews the process of inverse computation of the original PDDP algorithm with the introduction of the defect function $\mathbf{d}_{k+1}$ and modifies Equation (10) as follows

$$\delta \boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k + \delta \boldsymbol{x}_k, \boldsymbol{u}_k + \delta \boldsymbol{u}_k; \boldsymbol{\theta}_k + \delta \boldsymbol{\theta}_k) - \boldsymbol{x}_{k+1} \tag{11}$$

A second-order approximation of $v(\delta \boldsymbol{x}_k; \delta \boldsymbol{\theta})$ is required, namely

$$\begin{aligned} v(\delta \boldsymbol{x}_k; \delta \boldsymbol{\theta}) = & \frac{1}{2} (\delta \boldsymbol{x}_k)^\top \mathbf{S}_k \delta \boldsymbol{x}_k + \frac{1}{2} (\delta \boldsymbol{x}_k)^\top \mathbf{M}_k \delta \boldsymbol{\theta} \\ & + \frac{1}{2} (\delta \boldsymbol{\theta})^\top \mathbf{M}_k^\top \delta \boldsymbol{x}_k + \frac{1}{2} (\delta \boldsymbol{\theta})^\top \mathbf{G}_k \delta \boldsymbol{\theta} \\ & + \mathbf{s}_k^\top \delta \boldsymbol{x}_k + \mathbf{g}_k^\top \delta \boldsymbol{\theta} + z_k \end{aligned} \tag{12}$$

where $\mathbf{S}_k = v_{\boldsymbol{xx},k}$, $\mathbf{M}_k = v_{\boldsymbol{x\theta},k}$, $\mathbf{G}_k = v_{\boldsymbol{\theta\theta},k}$, $\mathbf{g}_k = v_{\boldsymbol{\theta},k}$ and z_k is the zero-order term. A second-order approximation of $\boldsymbol{f}$ in Equation (11) yields

$$\delta \boldsymbol{x}_{k+1} = \mathbf{d}_{k+1} + \Phi_k \delta \boldsymbol{x}_k + \beta_k \delta \boldsymbol{u}_k + \gamma_k \delta \boldsymbol{\theta} \tag{13}$$

Associative Equations (9) and (12) lead to the following

$$Q_k(\boldsymbol{x}_k, \boldsymbol{u}_k; \boldsymbol{\theta}) \approx \frac{1}{2} \begin{bmatrix} 1 \\ \delta \boldsymbol{x} \\ \delta \boldsymbol{u} \\ \delta \boldsymbol{\theta} \end{bmatrix}^\top \begin{bmatrix} 0 & Q_{\boldsymbol{x}}^\top & Q_{\boldsymbol{u}}^\top & Q_{\boldsymbol{\theta}}^\top \\ Q_{\boldsymbol{x}} & Q_{\boldsymbol{xx}} & Q_{\boldsymbol{xu}} & Q_{\boldsymbol{x\theta}} \\ Q_{\boldsymbol{u}} & Q_{\boldsymbol{ux}} & Q_{\boldsymbol{uu}} & Q_{\boldsymbol{u\theta}} \\ Q_{\boldsymbol{\theta}} & Q_{\boldsymbol{\theta x}} & Q_{\boldsymbol{\theta u}} & Q_{\boldsymbol{\theta\theta}} \end{bmatrix} \begin{bmatrix} 1 \\ \delta \boldsymbol{x} \\ \delta \boldsymbol{u} \\ \delta \boldsymbol{\theta} \end{bmatrix} \tag{14}$$

where

$$\begin{aligned}
Q_{\boldsymbol{x},k} &= l_{\boldsymbol{x},k} + \Phi_k^\top \mathbf{s}_{k+1} + \Phi_k^\top \mathbf{S}_{k+1} \mathbf{d}_{k+1} \\
Q_{\boldsymbol{u},k} &= l_{\boldsymbol{u},k} + \beta_k^\top \mathbf{s}_{k+1} + \beta_k^\top \mathbf{S}_{k+1} \mathbf{d}_{k+1} \\
Q_{\boldsymbol{\theta},k} &= l_{\boldsymbol{\theta},k} + \gamma_k^\top \mathbf{s}_{k+1} + \mathbf{g}_{k+1} + \gamma_k^\top \mathbf{S}_{k+1} \mathbf{d}_{k+1} + \mathbf{M}_{k+1}^\top \mathbf{d}_{k+1} \\
Q_{\boldsymbol{xx},k} &= l_{\boldsymbol{xx},k} + \Phi_k^\top \mathbf{S}_{k+1} \Phi_k \\
Q_{\boldsymbol{uu},k} &= l_{\boldsymbol{uu},k} + \beta_k^\top \mathbf{S}_{k+1} \beta_k \\
Q_{\boldsymbol{\theta\theta},k} &= l_{\boldsymbol{\theta\theta},k} + \mathbf{G}_{k+1} + \gamma_k^\top \mathbf{S}_{k+1} \gamma_k + \gamma_k^\top \mathbf{M}_{k+1} + \mathbf{M}_{k+1} \gamma_k \\
Q_{\boldsymbol{xu},k} &= l_{\boldsymbol{xu},k} + \Phi_k^\top \mathbf{S}_{k+1} \beta_k \\
Q_{\boldsymbol{x\theta},k} &= l_{\boldsymbol{x\theta},k} + \Phi_k^\top \mathbf{S}_{k+1} \gamma_k + \Phi_k^\top \mathbf{M}_{k+1} = Q_{\boldsymbol{\theta x},k}^\top \\
Q_{\boldsymbol{u\theta},k} &= l_{\boldsymbol{u\theta},k} + \beta_k^\top \mathbf{S}_{k+1} \gamma_k + \beta_k^\top \mathbf{M}_{k+1} = Q_{\boldsymbol{\theta u},k}^\top
\end{aligned} \tag{15}$$

The blue part of Equation (15) is where the MSPDDP differs from the original PDDP. And the expressions for $\mathbf{s}_k$, $\mathbf{g}_k$, $\mathbf{S}_k$, $\mathbf{M}_k$ and $\mathbf{G}_k$ are as follows

$$\begin{aligned}
\mathbf{s}_k &= Q_{\boldsymbol{x},k} - Q_{\boldsymbol{xu},k} Q_{\boldsymbol{uu},k}^{-1} Q_{\boldsymbol{u},k} \\
\mathbf{g}_k &= Q_{\boldsymbol{\theta},k} - Q_{\boldsymbol{\theta u},k} Q_{\boldsymbol{uu},k}^{-1} Q_{\boldsymbol{u},k} \\
\mathbf{S}_k &= Q_{\boldsymbol{xx},k} - Q_{\boldsymbol{xu},k} Q_{\boldsymbol{uu},k}^{-1} Q_{\boldsymbol{ux},k} \\
\mathbf{M}_k &= Q_{\boldsymbol{x\theta},k} - Q_{\boldsymbol{xu},k} Q_{\boldsymbol{uu},k}^{-1} Q_{\boldsymbol{u\theta},k} = V_{\boldsymbol{\theta x},k}^\top \\
\mathbf{G}_k &= Q_{\boldsymbol{\theta\theta},k} - Q_{\boldsymbol{\theta u},k} Q_{\boldsymbol{uu},k}^{-1} Q_{\boldsymbol{u\theta},k}
\end{aligned} \tag{16}$$

Remark 1. The states at waypoints are defined as shooting states, while the remaining states are the roll-out states. Similarly to the PDDP algorithm, the j-th flight time θ_j is updated at the time instant corresponding to the waypoints.

Remark 2. The derivatives $Q_{\boldsymbol{uu},k}$ and $V_{\boldsymbol{\theta\theta},w_j}$ must be positive in the entire MSPDDP algorithm. To address this problem, the authors in [22] utilize a method that focuses regularization on state variables:

$$\begin{aligned}
\mathbf{S}_k &= \mathbf{S}_k + \rho_\mu \mathbf{I}_n \\
\mathbf{G}_k &= \mathbf{G}_k + \rho_\nu \mathbf{I}
\end{aligned} \tag{17}$$

where ρ_v and ρ_q are regularization constants.

4.2 Merit Function

The MSPDDP algorithm requires the reduction of both the cost function and the defect function in the optimization process. However, the defect function increases the cost, creating potential conflicts between these two objectives. The merit function is able to combine two objectives into a single function, i.e.

$$M(\boldsymbol{X},\boldsymbol{U};\boldsymbol{\Theta}) = J(\boldsymbol{X},\boldsymbol{U};\boldsymbol{\Theta}) + \mu \left\|\mathbf{d}(\boldsymbol{X},\boldsymbol{U};\boldsymbol{\Theta})\right\|_p \tag{18}$$

in which the vector $\mathbf{d}(\boldsymbol{X},\boldsymbol{U};\boldsymbol{\Theta})$ denotes the set of defect functions, and $\mu > 0$ is the weight parameter which values the defect function and the cost. The condition for accepting the search step is

$$M(\hat{\boldsymbol{X}},\hat{\boldsymbol{U}};\hat{\boldsymbol{\Theta}}) - M(\boldsymbol{X},\boldsymbol{U};\boldsymbol{\Theta}) < 0 \tag{19}$$

5 Simulation Results

5.1 Mission Scenario

Since this paper focuses on the trajectory characteristics of the UAV and the trajectory optimization algorithm, the UAV dynamics model in this section is simplified as

$$\begin{cases} \dot{x} = V\cos\psi \\ \dot{y} = V\sin\psi \\ \dot{\psi} = \dfrac{a_y}{V} \end{cases} \tag{20}$$

in which ψ is the flight angle and a_y is the lateral acceleration, i.e. control input. We define the cost function as

$$\begin{aligned} J = {} & \frac{1}{2}(\boldsymbol{x}_N - \boldsymbol{x}_d)^{\mathrm{T}}\mathbf{Q}_f\,(\boldsymbol{x}_N - \boldsymbol{x}_d) \\ & + \sum_{k=0}^{N-1}\left[\frac{1}{2}\boldsymbol{u}_k^{\mathrm{T}}\mathbf{R}\boldsymbol{u}_k + \frac{1}{2}(\boldsymbol{x}_k - \boldsymbol{x}_d)^{\mathrm{T}}\mathbf{Q}_s(\boldsymbol{x}_k - \boldsymbol{x}_d)\right] \end{aligned} \tag{21}$$

in which $\boldsymbol{x}_d = [x_T, y_T, \psi_d]^T$ is the required terminal state vector of the UAV. $\mathbf{Q}_f$, $\mathbf{R}$ and $\mathbf{Q}_s$ are the weighted diagonal matrices of the cost function. Table 1 presents the parameter settings during the operation of the MSPDDP algorithm.

Table 1: Parameters of MSPDDP algorithm

Parameters	Values	Parameters	Values
Weighting matrix $\mathbf{Q}_i^f$	diag($10\mathbf{I}_3$)	LM parameter ρ_v	10^{-5}
Weighting matrix $\mathbf{R}_i$	1.0	LM parameter ρ_q	10^{-5}
State weighting matrix $\mathbf{Q}_i^s$	$\mathbf{0}$	Damping coefficient α_l	0.9
Weighting parameter μ	10.0	Stopping threshold ϵ	10^{-5}

5.2 Performance of the MSPDDP Algorithm

In this subsection, we perform numerical simulations of a low-speed and short-range UAV multiple waypoint reconnaissance mission to demonstrate the effectiveness of the MSPDDP algorithm in solving the spatial-temporal trajectory optimization problem.

To demonstrate the applicability of the MSPDDP algorithm to the UAV's flight at different angles and at different waypoints, we make the UAV adopt different flight angles to pass through six waypoints and reach the terminal position. Since the inherent dynamics of the UAV, we assume that the speed magnitude of the UAV is constant for the simplicity of the calculations. The mission in this subsection is noted as Scenario 1. Table 2 gives the settings of each parameter of the UAV in Scenario 1.

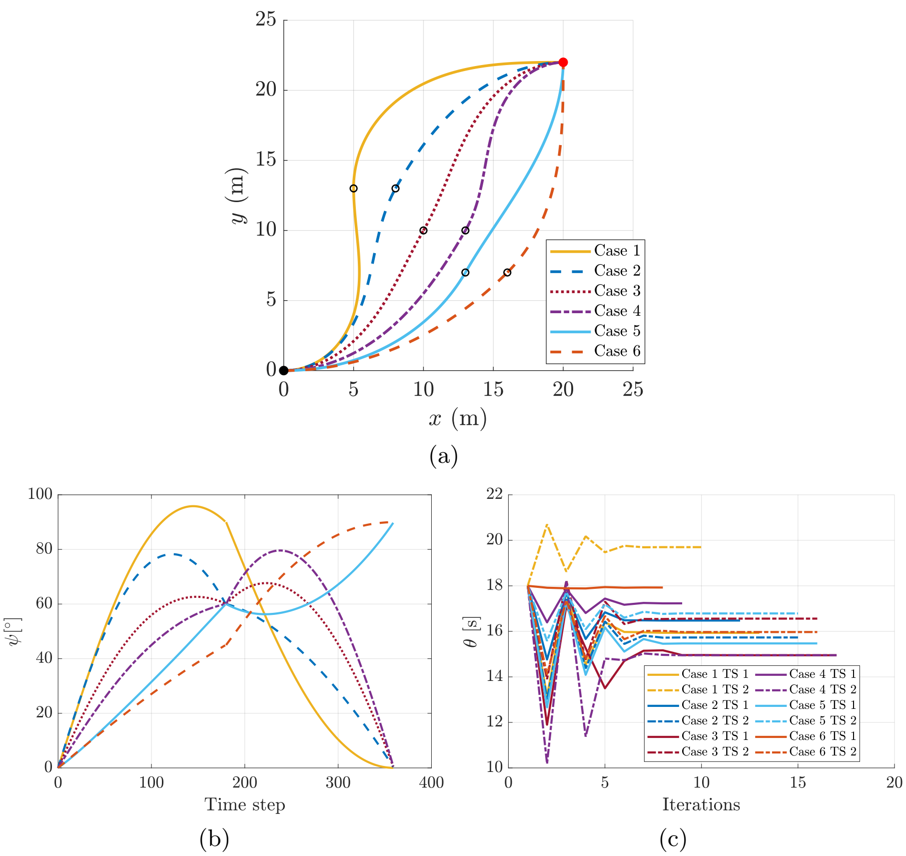

Fig. 2: Optimized trajectories of six cases (TS denotes the trajectory segment): (**a**) Flight trajectories. (**b**) Flight angle variations. (**c**) Terminal time results.

Table 2: Parameters of Scenario 1

Parameters	Values
UAV start point parameters $(x_0,\ y_0,\ \psi_0)$	$(0m,\ 0m,\ 0^\circ)$
Case 1-4 UAV terminal point parameters $(x_N,\ y_N,\ \psi_{N1})$	$(20m,\ 22m,\ 0^\circ)$
Case 5-6 UAV terminal point parameters $(x_N,\ y_N,\ \psi_{N2})$	$(20m,\ 22m,\ 90^\circ)$
Waypoint 1 parameters $\boldsymbol{p}_{w_1}$	$(5m,\ 13m,\ 90^\circ)$
Waypoint 2 parameters $\boldsymbol{p}_{w_1}$	$(8m,\ 13m,\ 60^\circ)$
Waypoint 3 parameters $\boldsymbol{p}_{w_1}$	$(10m,\ 10m,\ 60^\circ)$
Waypoint 4 parameters $\boldsymbol{p}_{w_1}$	$(13m,\ 10m,\ 60^\circ)$
Waypoint 5 parameters $\boldsymbol{p}_{w_1}$	$(13m,\ 7m,\ 60^\circ)$
Waypoint 6 parameters $\boldsymbol{p}_{w_1}$	$(16m,\ 7m,\ 45^\circ)$
UAV flight speed	$1.0m/s$
Lateral acceleration limitation $a_{y_{\max}}$	$30^\circ/s^2$
Initial solution guess for each trajectory flight time θ_j	$18.0s$
Flight time limitations for each segment $[t_{\min}, t_{\max}]$	$[5s, 38s]$

Figure 2 shows the results of the optimized trajectories of the UAV flying through six different waypoints at a specified flight angle. These include the optimized trajectories of the UAVs, flight angle variations, and flight time convergence. The simulation results indicate that the MSPDDP algorithm is capable of optimizing the trajectories of the UAV with multiple waypoints.

5.3 Comparison with MSDDP

This subsection compares the simulation with the MSDDP algorithm which cannot optimize flight time, denoted as Scenario 2. The simulation settings for MSDDP and MSPDDP are completely identical for the fairness of the comparision which are shown in Table 3.

The optimized trajectory results of the MSPDDP and fixed-time MSDDP algorithm for the UAV in the case of passing two waypoints are shown in Figure 3. The MSPDDP algorithm optimizes to smoother trajectories. The flight times for all three trajectories of MSPDDP are optimized to optimal values of 15.5034s, 16.6343s and 16.6591s, respectively. Each trajectory segment's flight time of MSDDP is identity 18s. Meanwhile, the defect function in the MSPDDP algorithm also gives better results in convergence. Considering the target position as a waypoint, the values of the defect function at all three waypoints are as shown in Table 4.

Comparison of the convergence results of the defect function of the MSDDP and MSPDDP algorithms shows that the MSPDDP algorithm has a less error at the waypoint. The trajectory results show the effectiveness of this proposed MSPDDP algorithm in a UAV free-time waypoint-trajectory optimization prob-

Table 3: Parameters of Scenario 2

Parameters	Values
UAV start point parameters $(x_0,\ y_0,\ \psi_0)$	$(0m,\ 0m,\ 0^{\circ})$
UAV terminal point parameters $(x_N,\ y_N,\ \psi_{N1})$	$(28m,\ 33m,\ 90^{\circ})$
Waypoint 1 parameters $\boldsymbol{p}_{w_1}$	$(8m,\ 12m,\ 60^{\circ})$
Waypoint 2 parameters $\boldsymbol{p}_{w_1}$	$(23m,\ 18m,\ 30^{\circ})$
UAV flight speed	$1.0m/s$
Lateral acceleration limitation $a_{y_{\max}}$	$30^{\circ}/s^2$
Initial solution guess for each trajectory flight time θ_j	$18.0s$
Flight time limitations for each segment $[t_{\min}, t_{\max}]$	$[5s, 38s]$

Table 4: Terminal defect function $(\mathbf{d}(x), \mathbf{d}(y), \mathbf{d}(\psi))$ of MSPDDP and MSDDP

Position	MSPDDP	MSDDP
Waypoint 1 $\boldsymbol{p}_{w_1}$	$(1.2\mathrm{e}^{-3}, -3.4\mathrm{e}^{-4}, 5.6\mathrm{e}^{-3})$	$(0.0026, 0.0029, 0.0114)$
Waypoint 2 $\boldsymbol{p}_{w_2}$	$(-2.0\mathrm{e}^{-4}, 8.6\mathrm{e}^{-4}, -5.7\mathrm{e}^{-3})$	$(0.0039, 0.0019, -0.0092)$
Terminal point $\boldsymbol{p}_{w_N}$	$(4.6\mathrm{e}^{-4}, -1.5\mathrm{e}^{-5}, 5.9\mathrm{e}^{-4})$	$(0.0012, 0.0028, 0.0061)$

lem. The algorithm is able to optimize the flight time of each trajectory segment while ensuring that the trajectory passes through multiple waypoints.

6 Conclusions and Future Work

This paper proposes a multiple shooting parameterized differential dynamic programming (MSPDDP) algorithm aiming at solving the UAV free-time waypoint-trajectory optimization problem in the battlefield reconnaissance mission of the UAV, augmenting the defect function into a parameterized action-value function, and parameterizing the flight time of each trajectory segment in order to dynamically optimize the flight time.

In this paper, we consider UAV reconnaissance scenarios and numerically simulate six different waypoints to verify the effectiveness of the proposed MSPDDP algorithm. In comparison with the MSDDP method which cannot optimize the flight time, the MSPDDP algorithm can optimize the flight times of the three trajectory segments to 15.5034s, 16.6343s, and 16.6591s, and the defect function is able to converge to lower values so the tolerances are smaller.

However, the paper has the following limitations. The MSPDDP algorithm is still unable to handle situations with obstacles, and the algorithm may fail to converge when the number of waypoints is large or when the flight angle between the start and terminal positions varies greatly. In our future work we

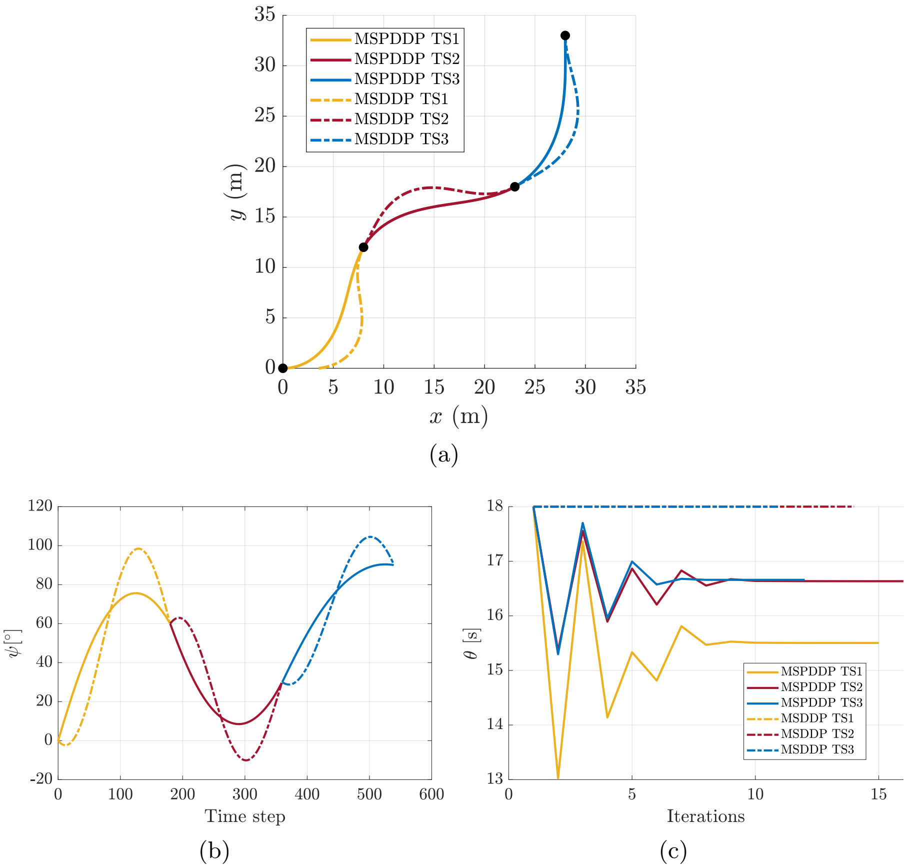

Fig. 3: Comparision for MSDDP and MSPDDP (TS refers to the trajectory segment): (**a**) Flight trajectories. (**b**) Flight angle variations. (**c**) Terminal time results.

will gradually solve these problems and perform numerical simulations in 3D scenarios using a UAV 6 DOF dynamics model.

References

1. Nex, F., Remondino, F.: UAV for 3D mapping applications: a review. J. Applied Geomatics, 6, 1–15 (2014)
2. Qi, J., Song, D., Shang, H., Wang, N., Hua, C., Wu, C., Qi, X., Han, J.: Search and Rescue Rotary-Wing UAV and Its Application to the Lushan Ms 7.0 Earthquake. J. Field Robotics, 33, 290–321 (2016)
3. Willems, J.: 1696: the birth of optimal control. In: Proceedings of 35th IEEE Conference on Decision and Control, pp. 1586-1587 (1996)

4. Euler, L.: Elementa calculi variationum. J. Novi commentarii academiae scientiarum Petropolitanae, 51–93 (1766)
5. Kirk, D. E.: Optimal Control Theory: An Introduction. J. Positively Aware the Monthly Journal of the Test Positive Aware Network, 23(2), 13–5 (2004)
6. Bellman, R., Pontryagin, L. S., Boltyanskii, V. G. B., et al.: Mathematical Theory of Optimal Processes. (1962)
7. He, S., Shin, H. S., Tsourdos, A.: Computational Guidance Using Sparse Gauss-Hermite Quadrature Differential Dynamic Programming. J. IFAC-PapersOnLine, (2019)
8. Bertsekas, D. P., Dimitrib.: Reinforcement Learning and Optimal Control. (2020)
9. Yakowitz, S. J.: Convergence rate analysis of the state increment dynamic programming method. J. Autom., 19, 53–60 (1983)
10. Manchester, Z., Kuindersma, S.: Derivative-free trajectory optimization with unscented dynamic programming. In: 2016 IEEE 55th Conference on Decision and Control (CDC), 3642–3647 (2016)
11. Tassa, Y., Mansard, N., Todorov, E.: Control-limited differential dynamic programming. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), 1168–1175 (2014)
12. Yakowitz, S. J.: The stagewise Kuhn-Tucker condition and differential dynamic programming. J. IEEE Trans. Autom. Control, 31, 25–30 (1986)
13. Xie, Z., Liu, C. K., Hauser, K. K.: Differential dynamic programming with nonlinear constraints. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), 695–702 (2017)
14. Li, H., Wensing, P. M.: Hybrid Systems Differential Dynamic Programming for Whole-Body Motion Planning of Legged Robots. J. IEEE Robotics and Automation Letters, 5, 5448–5455 (2020)
15. Lantoine, G., Russell, R. P.: A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 2: Application J. Journal of Optimization Theory and Applications, 154, 418–442 (2012)
16. Aoyama, Y., Boutselis, G. I., Patel, A., et al.: Constrained Differential Dynamic Programming Revisited. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), 9738–9744 (2020)
17. Oshin, A., Houghton, M. D., Acheson, M. J., et al.: Parameterized Differential Dynamic Programming. arXiv preprint (2022)
18. Giftthaler, M., Neunert, M., Stäuble, M., et al.: A Family of Iterative Gauss-Newton Shooting Methods for Nonlinear Optimal Control. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1–9 (2018)
19. Pellegrini, E., Russell, R. P.: A multiple-shooting differential dynamic programming algorithm. Part 1: Theory. J. Acta Astronautica, 170, 686–700 (2020)
20. Foehn, P., Romero, A., Scaramuzza, D.: Time-optimal planning for quadrotor waypoint flight. J. Science Robotics, 6 (2021)
21. Zheng, X., Xia, F., Lin, D., Jin, T., Su,W., He, S.: Constrained Parameterized Differential Dynamic Programming for Waypoint-Trajectory Optimization. J. Aerospace, 11, 420 (2024)
22. Tassa, Y., Erez, T., Todorov, E.: Synthesis and stabilization of complex behaviors through online trajectory optimization. In: Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4906–4913 (2012)