Superlinearly Convergent PMP-Based Algorithms for Nash-Coupled Nonlinear Discrete-Time Control

Xiaoqian Li, Zonglei Jing, Zhongjin Guo, Peijun Ju, and Shen Zhang

***Abstract*—This paper investigates the optimal control of discrete-time nonlinear systems with two strategically coupled control inputs arising from a Nash-type interaction. By augmenting the coupled inputs into a unified control vector and embedding the equilibrium conditions into a single performance index, the original game-based problem is reformulated as an augmented optimal control problem. Under standard regularity assumptions, we establish that the Nash stationarity conditions are locally equivalent to the first-order optimality conditions characterized by the Pontryagin maximum principle (PMP). Based on this reformulation, two PMP-based iterative algorithms are developed to solve the resulting forward-backward difference equations (FBDEs). The first algorithm exploits exact second-order information through the inversion of a regularized Hessian matrix, while the second constructs a Hessian-free recursive update that avoids explicit Hessian inversion. For both methods, a rigorous local convergence analysis is provided, showing that the effective linear contraction factor vanishes as the iteration horizon increases, leading to a superlinear-type local convergence behavior. Numerical experiments on representative nonlinear systems validate the theoretical findings and demonstrate improved convergence efficiency compared with classical gradient-based and Newton-type methods. The proposed theoretical framework and algorithms are particularly relevant to cooperative control problems in unmanned aerial vehicle (UAV) swarms, where multiple decision-makers interact through coupled dynamics and performance objectives.**

***Index Terms*—Nonlinear discrete-time optimal control, Nash equilibrium and game-theoretic control, Pontryagin maximum principle, multi-agent cooperative control**

I. Introduction

OPTIMAL control of nonlinear discrete-time systems is fundamental to modern control engineering and real-time decision-making applications [1–10], including cooperative control of unmanned aerial vehicle (UAV) swarms, networked multi-agent systems, and distributed energy systems. Compared with linear dynamics, nonlinear systems pose substantial analytical and computational challenges, which are further amplified in multi-input and multi-agent settings [6–12]. In such applications, control actions are often strategically coupled through shared system dynamics or mission-level performance objectives, giving rise to game-theoretic optimal control formulations that are particularly relevant for cooperative-competitive aerospace control and electronic decision systems.

The Pontryagin maximum principle (PMP) provides a cornerstone for nonlinear optimal control by characterizing optimal solutions via Hamiltonian systems and associated forward-backward difference equations (FBDEs) [13–16]. Despite its theoretical appeal, solving the resulting two-point boundary value problems remains challenging, particularly for discrete-time nonlinear systems with strong coupling and high dimensionality [16], as commonly encountered in cooperative multi-vehicle and UAV swarm control. To improve numerical tractability, several PMP-based algorithms explicitly exploit the FBDE structure. For example, Ref. [17] proposed a Newton-type framework that computes exact gradients and Hessian matrices from PMP conditions, achieving fast local convergence for single-agent nonlinear control problems. Recent developments have also explored accelerated gradient and Hessian computation in nonlinear optimal control [18] and interpreted constrained optimization as an optimal control problem [19]. However, such methods generally assume decoupled control channels and cannot be directly extended to strategically coupled inputs.

From a numerical standpoint, Newton-type methods are sensitive to Hessian conditioning, and may become unstable or divergent when the Hessian matrix is singular or ill-conditioned. To enhance robustness, a broad line of second-order optimization methods has been studied, including overview treatments of second-order derivative optimization, approximate Newton methods, and distributed quasi-Newton schemes [20–28]. Meanwhile, gradient-based methods rely heavily on step-size selection; although line search and adaptive strategies can improve stability [29–35], they often suffer from slow convergence in nonlinear or ill-conditioned problems.

Motivated by these limitations, increasing attention has been devoted to game-theoretic control of dynamic systems, particularly the computation of Nash equilibria in discrete-time and distributed settings [36–43]. These studies span multi-autonomous underwater vehicle (AUV) confrontation games, uncertain nonlinear systems, hybrid heterogeneous open multi-agent systems under denial of service (DoS) attacks, logical dynamic games, fixed-time and event-triggered Nash equilibrium seeking, and bilevel aggregative games. In such problems, multiple agents make simultaneous decisions while their objectives are mutually dependent, as in cooperative-

Manuscript received: 23 March 2026; revised: 10 April 2026; accepted: 17 April 2026. *(Corresponding author: Zonglei Jing.)*

Citation: X. Li, Z. Jing, Z. Guo, P. Ju, and S. Zhang, Superlinearly convergent PMP-based algorithms for Nash-coupled nonlinear discrete-time control, *Int. J. Intell. Control Syst.*, 2026, 31(2), 149–159.

Xiaoqian Li, Zhongjin Guo, Peijun Ju, and Shen Zhang are with School of Mathematics and Statistics, Taishan University, Taian 271000, China (e-mail: xiaoqianli0324@126.com; wuhanzhongjin@126.com; jpj615@163.com; szhang@tsu.edu.cn).

Zonglei Jing is with School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: raykr@buaa.edu.cn).

Digital Object Identifier 10.62678/IJICS202606.10352

competitive UAV swarm coordination and networked autonomous systems, rendering classical centralized optimal control inapplicable. Although several approaches have been proposed for related hierarchical and distributed decision-making problems [44–46], most existing results focus on linear dynamics or rely on restrictive assumptions. For nonlinear discrete-time systems with coupled performance indices, the systematic integration of Nash equilibrium conditions into PMP-based frameworks remains largely open.

In this paper, we study nonlinear discrete-time systems with two Nash-coupled control inputs. By embedding the equilibrium conditions into a unified performance index, the original Nash game is reformulated as an equivalent optimal control problem with an augmented control vector. Under suitable regularity assumptions, the reformulated problem admits a standard PMP characterization, leading to a set of FBDEs that encode both optimality and equilibrium conditions. Importantly, we show that under mild regularity assumptions, the Nash equilibrium conditions of the original two-player game are locally equivalent to the first-order optimality conditions of the augmented optimal control problem characterized by the Pontryagin maximum principle. This equivalence enables the direct application of PMP-based numerical algorithms to Nash-coupled control problems.

Building on this framework, we develop two superlinearly convergent algorithms for solving the resulting FBDEs. The first algorithm exploits exact second-order derivative information and achieves fast local convergence via Hessian inversion. The second algorithm avoids explicit Hessian inversion by constructing a recursively updated descent direction, while still retaining superlinear convergence. Rigorous convergence analysis is provided, and numerical experiments demonstrate the effectiveness of the proposed methods.

The main contributions of this paper are summarized as follows:

(1) A unified augmentation framework is developed to reformulate the Nash-coupled control inputs as a single optimal control variable, thereby enabling the application of the PMP.

(2) Two PMP-based algorithms are proposed, both of which exhibit locally superlinear convergence behavior, including a Hessian-free variant that improves numerical robustness.

(3) Nonlinear optimal control methods are extended to systems with strategically coupled inputs, which indicates potential applications to multi-agent and hierarchical control problems.

Notation Throughout the paper, $\mathbb{R}^n$ denotes the n-dimensional Euclidean space. $x_i \in \mathbb{R}^n$ denotes the system state at stage i, and $u_i, w_i \in \mathbb{R}^m$ denote two Nash-coupled control inputs. The augmented control variable is defined as $\Lambda_i = [u_i^{\mathrm{T}}, w_i^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{2m}$, and $\tau_i \in \mathbb{R}^{2m}$ denotes the auxiliary control variable that governs the update of Λ_i. The matrix $R \in \mathbb{R}^{2m\times 2m}$ is a symmetric positive definite weighting matrix used for regularization. The costate variable is denoted by $\zeta_i \in \mathbb{R}^{2m}$, and M represents the iteration horizon or time length. For a matrix A, A^{T} denotes its transpose, and I denotes the identity matrix. 0 denotes the zero vector. $\|\cdot\|$ represents the Euclidean norm for vectors and the induced spectral norm for matrices. The cost function is denoted by J_M, and all gradients and Hessian matrices are taken with respect to the augmented variable unless otherwise specified. The partial derivative of cost function is denoted by $\partial J_M / \partial x$, while $\nabla J_M(x)$ and $\nabla^2 J_M(x)$ represent its first-order derivative and second-order derivative, respectively. We use ∇J_M to denote gradient, and use $\partial J_M / \partial x$ for partial derivatives when needed.

II. Problem Formulation

Analytical solutions to nonlinear optimal control problems are generally unavailable, which motivates the development of efficient numerical algorithms. In this paper, we consider a class of discrete-time optimization problems with Nash-coupled control inputs, formulated as

$$\min_{\{u_i,w_i\}} J_M(x_0, u_i, w_i) \tag{1}$$

where u_i and w_i denote two control inputs with symmetric decision-making roles, and x_0 is the given initial condition. Such formulations naturally arise in decentralized and game-theoretic control systems, where multiple decision variables influence a common performance index.

Remark 1 Although Formula (1) is written as a centralized minimization, the term “Nash-coupled” is used to emphasize that the first-order stationarity conditions with respect to u_i and w_i form a coupled equilibrium-type system. Equivalently, one may view Formula (1) as the potential formulation of a two-player game, where both decision variables interact through a shared performance index.

Throughout this paper, we assume that J_M is twice continuously differentiable with respect to its arguments. Classical numerical optimization methods, such as gradient descent (GD) or Newton-type schemes, aim to compute stationary points of J_M by iteratively updating the decision variables. To facilitate algorithmic design and theoretical analysis, Formula (1) is first reformulated as a discrete-time optimal control problem.

A. Equivalent Optimal Control Representation

We introduce the following discrete-time dynamical system

$$\begin{aligned} J_M &= \sum_{i=0}^{M} \varPhi(x_i, u_i, w_i), \\ \text{s.t.,} \quad & x_{i+1} = \varGamma(x_i, u_i, w_i), \\ & i = 0, 1, \ldots, M \end{aligned} \tag{2}$$

where $\varPhi : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m \Longrightarrow \mathbb{R}$ denotes the running cost, $\varGamma : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m \Longrightarrow \mathbb{R}^n$ defines the system dynamics, $x_i \in \mathbb{R}^n$ is the system state at stage i, and x_0 is given.

Remark 2 The cost functional in Eq. (2) adopts a standard Lagrangian form. Under this formulation, each control input (u_i, w_i) influences both the immediate stage cost and future system evolution, which is essential for the application of optimal control techniques.

Remark 3 Formulas (1) and (2) are equivalent in the sense that they admit identical optimal control sequences. This equivalence allows the original Nash-coupled optimization problem to be addressed using tools from optimal control theory, particularly the Pontryagin maximum principle.

Remark 4 Regularity conditions for PMP The application of PMP in this paper relies on standard assumptions commonly adopted in discrete-time optimal control and differential games. Specifically, the system dynamics $\Gamma(\cdot)$ are assumed to be continuously differentiable and locally Lipschitz, the admissible control sets are convex, and the stage cost $\Phi(\cdot)$ is twice continuously differentiable. These conditions ensure the existence of well-defined adjoint variables and the validity of the first-order necessary conditions derived below.

The state trajectory generated by Eq. (2) can be expressed explicitly as

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix} = \begin{bmatrix} \Gamma(x_0,u_0,w_0) \\ \Gamma(x_1,u_1,w_1) \\ \vdots \\ \Gamma(x_{M-1},u_{M-1},w_{M-1}) \end{bmatrix} \tag{3}$$

which implies that each x_i is a function of the initial condition and past control inputs. Rather than solving Eq. (2) directly, we next construct an augmented optimal control formulation that facilitates efficient numerical computation.

B. Augmented Control Reformulation

To unify the Nash-coupled control inputs into a single control framework, we introduce the augmented variable

$$\Lambda_i \triangleq \begin{bmatrix} u_i \\ w_i \end{bmatrix} \in \mathbb{R}^{2m},$$

and define an auxiliary control input $\tau_i \in \mathbb{R}^{2m}$ governing the update of Λ_i. τ_i represents an incremental update direction for the unified control variable, analogous to a descent direction in optimization. Consider the following meta-level optimal control problem.

$$\begin{aligned} J_M &= \sum_{i=0}^{M-1}\left(J_M(x_0,\Lambda_i)+\frac{1}{2}\tau_i^{\mathrm{T}}R\tau_i\right)+J_M(x_0,\Lambda_M), \\ \text{s.t.,}\quad & \Lambda_{i+1}=\Lambda_i+\tau_i, \\ & i=0,1,\ldots,M-1, \\ & \Lambda_0=\eta \end{aligned} \tag{4}$$

where $R \in \mathbb{R}^{2m\times 2m}$ is a symmetric positive definite weighting matrix. It acts as a regularization term that balances update magnitude and stability, similar to trust-region or damping mechanisms in second-order optimization. The vector η denotes the initial estimate of the control variable.

The Hamiltonian associated with Eq. (4) is given by

$$H_i = J_M(x_0,\Lambda_i)+\frac{1}{2}\tau_i^{\mathrm{T}}R\tau_i+\zeta_{i+1}^{\mathrm{T}}(\Lambda_i+\tau_i) \tag{5}$$

where $\zeta_i \in \mathbb{R}^{2m}$ is the costate vector. Applying PMP yields the following forward-backward difference equations

$$\zeta_i = \zeta_{i+1}+\nabla J_M(x_0,\Lambda_i) \tag{6}$$

$$0 = R\tau_i+\zeta_{i+1} \tag{7}$$

Remark 5 Equation (6) describes the backward propagation of the costate, while Eq. (7) characterizes the optimal update direction τ_i. Together, these conditions define a structured FBDE system whose solution drives Λ_i toward a stationary point of J_M.

This augmented formulation transforms the original Nash-coupled problem into a classical optimal control problem with a unified control input. The resulting structure enables the systematic design of PMP-based algorithms with provable convergence properties. In particular, the block structure induced by R and the linear update dynamics of Λ_i play a central role in the development of the superlinearly convergent algorithms presented in the subsequent sections.

III. A Superlinear Optimization Algorithm and Its Convergence Analysis

It is well known that any optimal solution of a discrete-time optimal control problem must satisfy the associated forward-backward difference equations derived from the Pontryagin maximum principle. Accordingly, the optimal state and control trajectories of the augmented optimal control problem can be obtained by solving the coupled FBDE system in Eqs. (6) and (7). Although classical numerical schemes such as successive approximation can be applied, their convergence is typically slow due to the strong coupling between forward and backward dynamics.

Motivated by this limitation, we propose a superlinearly convergent iterative algorithm that exploits the special structure of the augmented problem. The key idea is to combine PMP-based optimality conditions with Taylor expansion and recursive approximation, leading to a computationally efficient update rule for the unified control variable.

A. Characterization of the Optimal Update

Lemma 1 Let Λ^* be a stationary point of Formula (1). Then Λ^* satisfies the following recursive update relation

$$\Lambda_{i+1}^* = \Lambda_i^*+\tau_i^* \tag{8}$$

where

$$\tau_i^* = -R^{-1}\sum_{j=i+1}^{M}\nabla J_M(x_0,\Lambda_j) \tag{9}$$

The proof follows directly from the PMP optimality conditions, as shown in Appendix A.

Substituting Eq. (9) into the state update equation yields

$$\Lambda_{i+1} = \Lambda_i - R^{-1}\sum_{j=i+1}^{M}\nabla J_M(x_0,\Lambda_j) \tag{10}$$

Remark 6 When $M=i+1$, Eq. (10) reduces to the implicit update

$$\Lambda_{i+1} = \Lambda_i - R^{-1}\nabla J_M(x_0,\Lambda_{i+1}).$$

Approximating $\nabla J_M(x_0,\Lambda_{i+1})$ by $\nabla J_M(x_0,\Lambda_i)$ yields a gradient descent step. Further applying a first-order Taylor expansion leads to

$$\Lambda_{i+1}-\Lambda_i = -(R+\nabla^2 J_M(x_0,\Lambda_i))^{-1}\nabla J_M(x_0,\Lambda_i),$$

which coincides with Newton's method when $R=0$. Thus, the proposed framework generalizes both gradient descent and Newton-type methods.

B. Algorithm Design

Based on the recursive structure in Eq. (10), we define the following superlinear algorithm. Algorithm 1 consists of a backward recursion for $\bar{g}_i$ and a forward update for Λ_i.

Algorithm 1 Superlinear optimization algorithm

Require: Initial point Λ_0 and cost function $J_M(x_0,\Lambda)$

Ensure: Final iterate Λ_M

```
1: for i = M−1 down to 0 do
2:     if i = M−1 then
3:         ḡ_i ← (R + ∇²J_M(x_0, Λ_i))^{-1} ∇J_M(x_0, Λ_i)
4:     else
5:         ḡ_i ← (R + ∇²J_M(x_0, Λ_i))^{-1} [∇J_M(x_0, Λ_i) + R ḡ_{i+1}]
6:     end if
7: end for
8: for i = 0 to M−1 do
9:     Λ_{i+1} ← Λ_i − ḡ_i
10: end for
```

For clarity, Algorithm 1 can be summarized as follows

$$\Lambda_{i+1} = \Lambda_i - \bar{g}_i \tag{11}$$

$$\bar{g}_i = (R+\nabla^2 J_M(x_0,\Lambda_i))^{-1}[\nabla J_M(x_0,\Lambda_i)+R\bar{g}_{i+1}] \tag{12}$$

with terminal condition

$$\bar{g}_{M-1} = (R+\nabla^2 J_M(x_0,\Lambda_{M-1}))^{-1}\nabla J_M(x_0,\Lambda_{M-1}) \tag{13}$$

To facilitate a clearer presentation of the convergence analysis for Algorithm 1, we first introduce the following lemma.

Lemma 2 The update relation in Eq. (10) can be approximated by

$$\Lambda_{i+1} = \Lambda_i - g_i \tag{14}$$

$$\begin{aligned} g_i = {} & (R+\nabla^2 J_M(x_0,\Lambda_i))^{-1}[\nabla J_M(x_0,\Lambda_i)+ \\ & \sum_{j=i+1}^{M-1}(\nabla J_M(x_0,\Lambda_j)-\nabla^2 J_M(x_0,\Lambda_j)g_j)], \\ g_{M-1} = {} & (R+\nabla^2 J_M(x_0,\Lambda_{M-1}))^{-1}\nabla J_M(x_0,\Lambda_{M-1}) \end{aligned} \tag{15}$$

Proof of Lemma 2 is detailed in Appendix B.

C. Closed-Form Representation of the Backward Recursion

Before analyzing the local convergence behavior of Algorithm 1, we first derive a compact closed-form expression for the backward recursion in Eq. (12). This representation plays a key role in the subsequent convergence analysis.

The main technical idea is to adopt a local approximation by evaluating all higher-index quantities Λ_j ($j \geqslant i+1$) at the common reference point Λ_i. This approximation does not affect the local convergence order, as all higher-order terms vanish in a sufficiently small neighborhood of the stationary point.

Here, to simplify the description, we define

$$A_i \triangleq (R+\nabla^2 J_M(x_0,\Lambda_i))^{-1}, \qquad B_i \triangleq A_i R.$$

Under this approximation, the terminal condition in Eq. (13) becomes

$$\bar{g}_{M-1}(\Lambda_i) = A_i \nabla J_M(x_0,\Lambda_i).$$

Applying the recursion in Eq. (12) yields

$$\begin{aligned} \bar{g}_{M-2}(\Lambda_i) &= A_i[\nabla J_M(x_0,\Lambda_i)+R\bar{g}_{M-1}(\Lambda_i)] = \\ & A_i\nabla J_M(x_0,\Lambda_i)+B_i\bar{g}_{M-1}(\Lambda_i), \\ \bar{g}_{M-3}(\Lambda_i) &= A_i[\nabla J_M(x_0,\Lambda_i)+R\bar{g}_{M-2}(\Lambda_i)] = \\ & A_i\nabla J_M(x_0,\Lambda_i)+B_i\bar{g}_{M-2}(\Lambda_i). \end{aligned}$$

By backward induction, for any $k = i, i+1, \ldots, M-1$, the recursion admits the compact form

$$\begin{aligned} \bar{g}_k(\Lambda_i) &= A_i\nabla J_M(x_0,\Lambda_i)+B_i\bar{g}_{k+1}(\Lambda_i), \\ \bar{g}_{M-1}(\Lambda_i) &= A_i\nabla J_M(x_0,\Lambda_i) \end{aligned} \tag{16}$$

Iterating Eq. (16) yields the finite Neumann-series expansion, which is shown as

$$\bar{g}_i(\Lambda_i) = \sum_{j=0}^{M-i-1} B_i^j A_i \nabla J_M(x_0,\Lambda_i).$$

Since

$$\begin{aligned} I - B_i = {} & I-(R+\nabla^2 J_M(x_0,\Lambda_i))^{-1}R = \\ & (R+\nabla^2 J_M(x_0,\Lambda_i))^{-1}\nabla^2 J_M(x_0,\Lambda_i), \end{aligned}$$

we obtain

$$\begin{aligned} \bar{g}_i(\Lambda_i) = {} & [I-((R+\nabla^2 J_M(x_0,\Lambda_i))^{-1}R)^{M-i}]\times \\ & (\nabla^2 J_M(x_0,\Lambda_i))^{-1}\nabla J_M(x_0,\Lambda_i) \end{aligned} \tag{17}$$

and consequently

$$\begin{aligned} \Lambda_{i+1} = {} & \Lambda_i - [I-((R+\nabla^2 J_M(x_0,\Lambda_i))^{-1}R)^{M-i}]\times \\ & (\nabla^2 J_M(x_0,\Lambda_i))^{-1}\nabla J_M(x_0,\Lambda_i) \end{aligned} \tag{18}$$

Equations (17) and (18) provide an explicit representation of Algorithm 1 and will serve as the starting point for the local convergence analysis presented in the next subsection.

D. Convergence Analysis

We now analyze the local convergence behavior of Algorithm 1.

Lemma 3 Suppose $\nabla J_M(x_0,\Lambda_*) = 0$ and $\nabla^2 J_M(x_0,\Lambda_*)$ is positive definite. Then, for sufficiently large M, the sequence $\{\Lambda_i\}$ generated by Algorithm 1 satisfies

$$\|\Lambda_{i+1}-\Lambda_*\| \leqslant \|(R+\nabla^2 J_M(x_0,\Lambda_*))^{-1}R\|^{M-i}\|\Lambda_i-\Lambda_*\| \tag{19}$$

Proof of Lemma 3 is detailed in Appendix C.

We now establish a refined local error estimate that characterizes the superlinear behavior of Algorithm 1.

Theorem 1 Under standard regularity conditions ensuring the validity of PMP and local twice continuous differentiability of J_M, assume that $\nabla J_M(x_0,\Lambda_*) = 0$ and $\nabla^2 J_M(x_0,\Lambda_*)$ is positive definite. Then the sequence generated by Algorithm 1 satisfies

$$\begin{aligned} \|\Lambda_{i+1}-\Lambda_*\| \leqslant {} & \|(R+\nabla^2 J_M(x_0,\Lambda_*))^{-1}R\|^{M-i}\|\Lambda_i-\Lambda_*\|+ \\ & \|\bar{g}_i''(\Lambda_*)\|\|\Lambda_i-\Lambda_*\|^2 \end{aligned} \tag{20}$$

where $\|(R+\nabla^2 J_M)^{-1}R\| < 1$. Consequently, for sufficiently large M, Algorithm 1 exhibits local superlinear convergence.

The result follows from a second-order Taylor expansion of

$\bar{g}_i(\Lambda)$ around Λ_* and the fact that the linear term vanishes asymptotically as M increases. The complete derivation is provided in the Appendix D.

Remark 7 Formula (20) consists of two parts: a vanishing linear term $\|(R+\nabla^2 J_M)^{-1}R\|^{M-i}\|\Lambda_i-\Lambda^*\|$ and a second-order error term. As $M\to\infty$, the linear coefficient tends to zero, and the quadratic term dominates, which implies superlinear convergence.

IV. A Hessian-Free Superlinear Optimization Algorithm

This section presents a Hessian-inversion-free variant of the superlinear algorithm developed in Section III, aimed at improving numerical robustness and scalability. The primary computational cost in the superlinear update arises from evaluating the inverse of the matrix $R+\nabla^2 J_M(x_0,\Lambda_i)$. To simplify notation, we define

$$\Xi := (R+\nabla^2 J_M(x_0,\Lambda_i))^{-1}.$$

The resulting scheme avoids repeated explicit inversion of the regularized Hessian matrix while retaining the recursive structure needed for superlinear local convergence.

Then we can rewrite

$$\begin{aligned}&(R+\nabla^2 J_M(x_0,\Lambda_i))^{-1}R=\\&I-(R+\nabla^2 J_M(x_0,\Lambda_i))^{-1}\nabla^2 J_M(x_0,\Lambda_i)=\\&I-\Xi\nabla^2 J_M(x_0,\Lambda_i).\end{aligned}$$

Consequently, the superlinear update in Eq. (18) can be expressed equivalently as

$$\begin{aligned}\Lambda_{i+1}=\Lambda_i-[I-(I-\Xi\nabla^2 J_M(x_0,\Lambda_i))^{M-i}]\times\\(\nabla^2 J_M(x_0,\Lambda_i))^{-1}\nabla J_M(x_0,\Lambda_i)\end{aligned}\tag{21}$$

A. Hessian-Free Algorithm

The Hessian-free iterative scheme replaces explicit Hessian inversion with the matrix-vector operator Ξ, thereby avoiding repeated inversion of $R+\nabla^2 J_M$ while retaining superlinear convergence. The scheme is described in Algorithm 2.

$$\Lambda_{i+1}=\Lambda_i-\hat{g}_i\tag{22}$$

$$\hat{g}_i=\Xi\nabla J_M(x_0,\Lambda_i)+(I-\Xi\nabla^2 J_M(x_0,\Lambda_i))\hat{g}_{i+1}\tag{23}$$

$$\hat{g}_{M-1}=\Xi\nabla J_M(x_0,\Lambda_{M-1})\tag{24}$$

Proof of Algorithm 2 is shown in Appendix E.

B. Convergence Analysis

Theorem 2 Assume that $\nabla^2 J_M(x_0,\Lambda_i)$ is positive definite and $\nabla J_M(x_0,\Lambda_i)=0$ at the stationary point Λ_*. If the spectral radius satisfies

$$\rho(I-\Xi\nabla^2 J_M(x_0,\Lambda_*))<1,$$

then the iterative sequence $\{\Lambda_i\}$ generated by Eq. (21) converges superlinearly to Λ_*

$$\|\Lambda_{i+1}-\Lambda_*\|\leqslant\rho(I-\Xi\nabla^2 J_M(x_0,\Lambda_*))^{M-i}\|\Lambda_i-\Lambda_*\|.$$

Consequently, as M increases, the effective linear contraction factor $\rho(I-\Xi\nabla^2 J_M(\Lambda_*))^{M-i}$ vanishes, and the higher-order term dominates, which yields a superlinear convergence behavior in the same spirit as Theorem 1.

Algorithm 2 Hessian-free superlinear optimization algorithm

Require: Initial point Λ_0, cost function $J_M(x_0,\Lambda)$, and operator Ξ

Ensure: Final iterate Λ_M

```
for i = M−1 down to 0 do
    if i = M−1 then
        ĝ_i ← Ξ∇J_M(x_0, Λ_i)
    else
        ĝ_i ← Ξ∇J_M(x_0, Λ_i) + (I − Ξ∇²J_M(x_0, Λ_i)) ĝ_{i+1}
    end if
end for
for i = 0 to M−1 do
    Λ_{i+1} ← Λ_i − ĝ_i
end for
return Λ_M
```

Proof From the linearization of the Hessian-free update in Eq. (21), we have

$$\begin{aligned}\hat{g}_i(\Lambda_i)=[I-(I-\Xi\nabla^2 J_M(x_0,\Lambda_i))^{M-i}]\times\\(\nabla^2 J_M(x_0,\Lambda_i))^{-1}\nabla J_M(x_0,\Lambda_i),\end{aligned}$$

when $\Lambda_i=\Lambda_*$, $\nabla J_M(x_0,\Lambda_*)=0$, $\hat{g}_i(\Lambda_*)=0$, and $\hat{g}_i'(\Lambda_*)=I-(I-\Xi\nabla^2 J_M(x_0,\Lambda_i))^{M-i}$.

Thus

$$\begin{aligned}\Lambda_{i+1}-\Lambda_*=\Lambda_i-\hat{g}_i(\Lambda_*)-\Lambda_*=\\&\Lambda_i-\Lambda_*-[\hat{g}_i(\Lambda_*)+\hat{g}_i'(\Lambda_*)(\Lambda_i-\Lambda_*)+\\&o(|\Lambda_i-\Lambda_*|)]\approx\\&\Lambda_i-\Lambda_*-\hat{g}_i'(\Lambda_*)(\Lambda_i-\Lambda_*)=\\&(I-\hat{g}_i'(\Lambda_*))(\Lambda_i-\Lambda_*)=\\&(I-\Xi\nabla^2 J_M(x_0,\Lambda^*))^{M-i}(\Lambda_i-\Lambda^*).\end{aligned}$$

Under the spectral radius condition $\rho(I-\Xi\nabla^2 J_M(x_0,\Lambda_*))<1$, the iteration is a contraction for each fixed M, and the contraction factor vanishes as M increases, leading to an accelerated (superlinear-type) local convergence behavior.

Remark 8 Although the augmented control dimension grows linearly with the horizon length M, the inclusion of the regularization matrix R ensures uniform positive definiteness of the approximate Hessian matrix. This prevents severe ill-conditioning in practice and guarantees that the spectral radius condition is satisfied for a suitably chosen Ξ.

This structure also enables the application of the proposed algorithms to trajectory optimization of nonlinear dynamical systems, which will be illustrated through numerical simulations in Section V.

V. Numerical Experiment

All numerical experiments were conducted using Python 3.11 on a desktop computer equipped with an Intel Core i9-13900K processor (24 cores and 32 threads) operating at 3.0–5.8 GHz. No GPU acceleration or parallelization was employed. All algorithms were implemented using native NumPy and executed in a single-threaded environment to ensure consistency of timing and convergence comparisons.

Unless otherwise specified, a default iteration horizon $M = 30$ was used in all experiments. The stopping criterion was set as $\|\Lambda_k - \Lambda^*\| < 10^{-6}$ or until the maximum iteration number was reached. All methods were initialized from the same starting point within each comparative experiment to ensure fairness. Any experiment-specific initialization or regularization settings are explicitly stated in the corresponding subsection.

To provide a comprehensive evaluation, the experiments are organized to examine convergence efficiency, robustness under ill-conditioned settings, scalability in high-dimensional problems, and sensitivity to the control weighting matrix, with comparisons to gradient-based and Newton-type methods.

A. Fast Convergence

The fast convergence performance of the proposed algorithms is demonstrated by considering a synthetic Nash-coupled nonlinear cost function, which serves as a representative abstraction of coupled decision-making in cooperative UAV swarm control. The cost function is defined as

$$\phi(x,u,w) = (1-x)^2 + 100(x-u)^2 + 100(w-\sin(u))^2,$$

where $x \in \mathbb{R}$ is fixed at $x_0 = 1.5$, and the optimization is performed over the coupled inputs $\Lambda = [u,w]^{\mathrm{T}} \in \mathbb{R}^2$. Here, the two control variables u and w may be interpreted as interacting control or coordination channels associated with a UAV agent, while x represents a shared system state or mission-related variable. The function exhibits strong convexity in w and mild nonlinearity in u, with non-separable coupling introduced through the sine term.

The convergence behavior of the two proposed methods—Algorithm 1 (Newton-type with regularized inverse) and Algorithm 2 (recursive form avoiding matrix inversion)—is compared with that of the classical gradient descent method. All algorithms are initialized from the same point $\Lambda_0 = [0.5, 1.0]^{\mathrm{T}}$, and the regularization matrix is fixed as $R = I$. The convergence is evaluated by the distance $\|\Lambda_k - \Lambda^*\|$, referred to as the relative error, where Λ^* denotes the converged solution obtained by Algorithm 1. All sequences are padded to 30 iterations for fair comparison. This experiment directly compares the proposed methods with a classical first-order baseline and illustrates the benefit of incorporating second-order curvature information.

Fig. 1 illustrates the corresponding results, highlighting that Algorithm 1 achieves rapid convergence within just 7 iterations, significantly outperforming gradient descent. Algorithm 2, while slightly slower due to its recursive formulation, still exhibits a comparable rate and quickly reaches a low residual plateau. In contrast, GD requires more than 10 steps to attain comparable accuracy and stagnates shortly thereafter due to its fixed step-size limitation.

The results show that the proposed methods exploit curvature information more effectively than the first-order baseline. In particular, Algorithm 1 converges within a few iterations, while Algorithm 2 achieves comparable accuracy with lower per-iteration complexity.

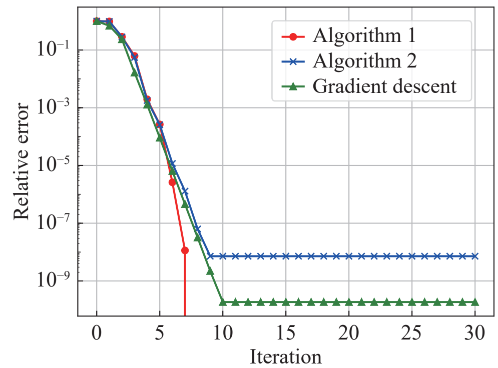

Fig. 1 Convergence comparison on a synthetic Nash-coupled function.

It is worth noting that classical gradient-based approaches, such as those surveyed by Ruder [47], often require careful tuning of step sizes or rely on momentum heuristics [30] to avoid stagnation. In contrast, our algorithms exhibit robust and consistent convergence behavior without requiring extensive hyperparameter adjustments, as also observed in adaptive methods like hypergradient descent [48].

B. Wide Applicability

The robustness of the proposed method is demonstrated in challenging cases where traditional second-order methods fail. Such scenarios frequently arise in practical control applications, including UAV swarm coordination, where different control channels may exhibit vastly different sensitivities and actuation scales. Specifically, we consider an extreme flat-direction optimization problem of the form

$$\phi(u,w) = 10^{-8}(u-1)^2 + (w-2)^2,$$

which is strongly convex in w but nearly flat in u. This structure reflects situations where one control direction is weakly penalized or poorly observable, while another dominates the system response. The corresponding Hessian matrix is highly ill-conditioned

$$\nabla^2\phi = \begin{bmatrix} 2\times 10^{-8} & 0 \\ 0 & 2 \end{bmatrix}.$$

A condition number exceeding 10^8 renders the Newton direction numerically unreliable along the nearly flat direction.

Algorithm 1, which introduces a regularized inverse via $R + \nabla^2\phi$, is compared against the classical Newton method. Both methods are initialized at $\Lambda_0 = [0,0]^{\mathrm{T}}$, and a regularization matrix $R = I$ is employed. The convergence is evaluated using the relative error $\|\Lambda_k - \Lambda^*\|$, where Λ^* denotes the final iterate of Algorithm 1.

Unlike traditional Newton methods that degrade under extreme conditioning, as discussed by Dembo et al. [24], the proposed regularization-based strategy ensures numerical stability. Although modified Newton variants [25] attempt to alleviate singularity issues, they often lack a structured integration into coupled control frameworks, which is critical in multi-input decision-making problems such as cooperative UAV control.

Fig. 2 presents the results, revealing that Newton's method stagnates immediately, failing to reduce the error even after 30 steps due to the dominance of the nearly singular direction in the Hessian matrix. In contrast, Algorithm 1 effectively regularizes the curvature and achieves full convergence within 14 iterations. This confirms the proposed method's superior stability and broad applicability, particularly in flat, degenerate, or ill-conditioned optimization landscapes commonly encountered in practical multi-agent control systems. This comparison further emphasizes the robustness of the proposed method relative to classical Newton-type approaches under ill-conditioned scenarios.

C. Efficiency under High-Dimensional Settings

In addition to convergence performance, we evaluate the computational efficiency of the proposed algorithms with respect to problem dimension. High-dimensional optimization problems naturally arise in practical control applications, such as UAV swarm coordination, where multiple agents, states, and control channels are jointly optimized. Specifically, we consider a family of synthetic quadratic optimization problems of the form

$$\min_{\Lambda} \ \phi(\Lambda) = \frac{1}{2}\Lambda^{\mathrm{T}} A \Lambda + b^{\mathrm{T}}\Lambda,$$

where $A \in \mathbb{R}^{n\times n}$ is a randomly generated symmetric positive definite matrix and $b \in \mathbb{R}^n$. For each dimension $n \in \{10, 50, \ldots, 1000\}$, we measure the average runtime per iteration over five steps for both Algorithms 1 and 2.

As depicted in Fig. 3, the runtime of Algorithm 1 increases rapidly with the problem dimension, exhibiting approximately cubic growth due to repeated matrix inversions in each iteration. In contrast, for this quadratic test problem, Algorithm 2 can precompute a single inverse and then perform only recursive updates thereafter, leading to substantially better scalability. For instance, at $n = 1000$, Algorithm 2 achieves a runtime reduction of over 45% compared to Algorithm 1.

These results confirm that while Algorithm 1 provides superior convergence performance in low-to-moderate dimensions, Algorithm 2 is significantly more efficient in high-dimensional settings. Such scalability advantages are particularly important for large-scale and real-time implementations, including cooperative control and decision-making in UAV swarms with many degrees of freedom. This experiment highlights the scalability advantage of the Hessian-free formulation compared to the Hessian-based implementation.

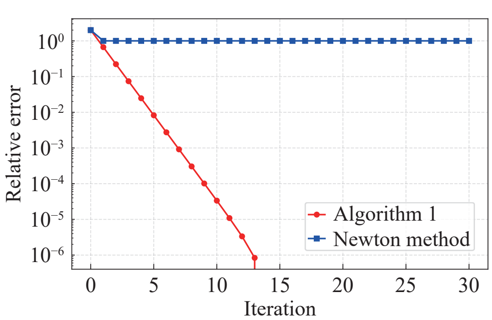

Fig. 2 Comparison in a flat-direction problem. Algorithm 1 converges reliably, and Newton method stagnates due to near-singular Hessian matrix.

D. Effect of Control Weight R on Convergence

We further investigate how the control weighting matrix R influences the convergence performance of the proposed algorithms. This factor is crucial in optimal control formulations, where R balances the trade-off between control effort and stability of convergence.

We use the same nonlinear cost function introduced and evaluate both Algorithms 1 and 2 under a range of R values

$$R \in \{0.1I, 10I, 100I, 300I, 500I\},$$

with the initial state fixed at $x_0 = 1.5$ and $\Lambda_0 = [0.5, 1.0]^{\mathrm{T}}$. The convergence is measured in terms of $\|\Lambda_k - \Lambda^*\|$, where Λ^* denotes the final iterate of Algorithm 1 under each R.

Fig. 4 summarizes the results and reveals that both algorithms benefit from small to moderate values of R, promoting aggressive descent without excessive regularization. In particular, Algorithm 1 achieves rapid convergence when $R \leqslant 10I$, while large values of R (e.g., $R = 300I$ and $R = 500I$) lead to slower updates and delayed convergence.

Algorithm 2 exhibits greater sensitivity to R, with its convergence curve becoming oscillatory or stalled for large values of R. This is attributed to the recursive form being more susceptible to scaling imbalance and accumulated numerical errors when the step direction is heavily regularized.

These findings underscore the importance of properly tuning R to strike a balance between convergence rate and numerical stability. In practice, $R \in [I, 10I]$ serves as a reliable and effective default across a range of problem settings.

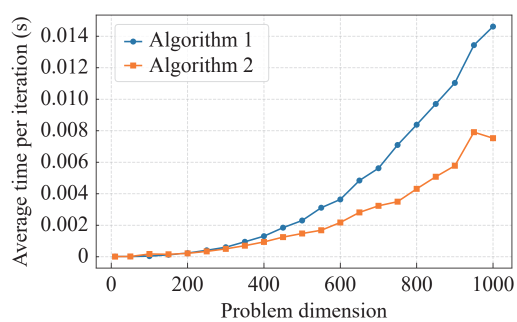

Fig. 3 Average per-iteration runtime for Algorithms 1 and 2 under an increasing problem dimension.

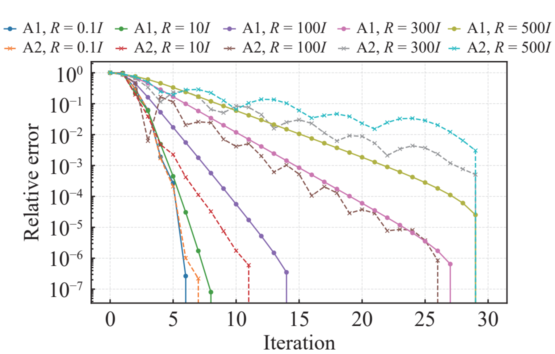

Fig. 4 Convergence of Algorithms 1 and 2 under different control weightings $R \in \{0.1I, 10I, 100I, 300I, 500I\}$. Smaller R yields faster descent, while large R slows both methods, with Algorithm 2 exhibiting more sensitivity. A1 stands for Algorithm 1 and A2 stands for Algorithm 2.

This sensitivity analysis further illustrates how the control weighting matrix influences convergence behavior across different algorithmic structures.

VI. Conclusion

This paper investigates a class of discrete-time nonlinear systems with two strategically coupled control inputs, where the underlying interaction follows a Nash equilibrium structure. By augmenting the control variables into a unified higher-dimensional vector and reformulating the performance index accordingly, the original problem is reformulated as an equivalent optimal-control problem that can be analyzed via PMP. The resulting FBDEs are then solved using two iterative algorithms with local superlinear convergence. The first algorithm incorporates complete second-order derivative information and achieves rapid convergence by using Hessian inversion. In contrast, the second algorithm circumvents the inversion operation and utilizes a recursively constructed descent direction while preserving superlinear convergence guarantees. For both algorithms, theoretical convergence analysis has been rigorously established, and their performance has been verified through numerical experiments, demonstrating superior convergence rates compared to classical approaches.

The proposed framework provides a structured approach for extending PMP-based optimization to Nash-coupled systems, with potential applicability to multi-player dynamic games, hierarchical control problems, and cooperative control scenarios in UAV swarms, where multiple agents interact through coupled dynamics and shared performance objectives. In addition, the framework is applicable to data-driven control paradigms, such as model-based reinforcement learning. Future research will focus on scalability in large-scale systems, robustness under model uncertainty, and real-time applicability in embedded control platforms, which are critical requirements for autonomous multi-agent systems.

Appendix A
Proof of Lemma 1

Proof By applying the Pontryagin maximum principle to the augmented optimal control problem in Eq. (4), the necessary optimality conditions are characterized by the forward-backward difference equations, shown in Eqs. (6) and (7).

From Eq. (6), the adjoint variable satisfies

$$\zeta_{M-1} = \zeta_M + \nabla J_M(x_0, \Lambda_{M-1}) = \sum_{j=M-1}^{M} \nabla J_M(x_0, \Lambda_j),$$

and by backward induction

$$\zeta_i = \sum_{j=i}^{M} \nabla J_M(x_0, \Lambda_j).$$

Substituting the above expression into Eq. (7) yields

$$\tau_i = -R^{-1}\zeta_{i+1} = -R^{-1} \sum_{j=i+1}^{M} \nabla J_M(x_0, \Lambda_j),$$

which completes the proof. ■

Appendix B
Proof of Lemma 2

This proof is established via backward mathematical induction on the index i, combined with first-order Taylor expansions of the gradient $\nabla J_M(x_0, \Lambda_j)$.

Proof

Step 1: Base case ($i = M-1$)

For $i = M-1$, consider Eq. (4)

$$\Lambda_M = \Lambda_{M-1} + \tau_{M-1} = \Lambda_{M-1} - R^{-1}\zeta_M = \Lambda_{M-1} - R^{-1}\nabla J_M(x_0, \Lambda_M).$$

Applying a first-order Taylor expansion of $\nabla J_M(x_0, \Lambda_M)$ around Λ_{M-1} yields

$$\nabla J_M(x_0, \Lambda_M) \approx \nabla J_M(x_0, \Lambda_{M-1}) + \nabla^2 J_M(x_0, \Lambda_{M-1})(\Lambda_M - \Lambda_{M-1}).$$

Substituting this approximation and rearranging terms lead to

$$\Lambda_M - \Lambda_{M-1} = -(R + \nabla^2 J_M(x_0, \Lambda_{M-1}))^{-1} \nabla J_M(x_0, \Lambda_{M-1}),$$

which implies

$$\Lambda_M = \Lambda_{M-1} - g_{M-1},$$

where

$$g_{M-1} = (R + \nabla^2 J_M(x_0, \Lambda_{M-1}))^{-1} \nabla J_M(x_0, \Lambda_{M-1}).$$

Step 2: Induction hypothesis

Assume that for all indices k satisfying $i+1 \leqslant k \leqslant M-1$, the update relation holds

$$\Lambda_{k+1} = \Lambda_k - g_k,$$

where g_k is given by Eq. (15).

Step 3: Induction step

We now show that the relation also holds for index i. Starting from Eq. (10), we expand $\nabla J_M(x_0, \Lambda_j)$ for all $j > i$ using first-order Taylor expansions around Λ_{j-1} and substitute the induction hypothesis $\Lambda_j - \Lambda_{j-1} = -g_{j-1}$. Neglecting higher-order terms yields

$$\Lambda_{i+1} \approx \Lambda_i - R^{-1}[\nabla J_M(x_0, \Lambda_i) + \sum_{j=i+1}^{M-1} (\nabla J_M(x_0, \Lambda_j) - \nabla^2 J_M(x_0, \Lambda_j) g_j)].$$

Rearranging the above expression gives

$$\Lambda_{i+1} = \Lambda_i - g_i \tag{A1}$$

where

$$g_i = (R + \nabla^2 J_M(x_0, \Lambda_i))^{-1} \times \left[\nabla J_M(x_0, \Lambda_i) + \sum_{j=i+1}^{M-1} (\nabla J_M(x_0, \Lambda_j) - \nabla^2 J_M(x_0, \Lambda_j) g_j)\right] \tag{A2}$$

This completes the backward induction and establishes that Eqs. (14) and (15) provide valid first-order approximations to the update relation in Eq. (10) for all indices i. ■

Appendix C
Proof of Lemma 3

This section provides the detailed derivative analysis supporting Lemma 2.

Proof By differentiating Eqs. (12) and (13) with respect to Λ_i, we obtain

$$\bar{g}_i'(\Lambda_i) = (R+\nabla^2 J_M(x_0,\Lambda_i))^{-1}[\nabla^2 J_M(x_0,\Lambda_i)+ R\bar{g}_{i+1}'(\Lambda_i)] - (R+\nabla^2 J_M(x_0,\Lambda_i))^{-2}\times J_M'''(x_0,\Lambda_i)[\nabla J_M(x_0,\Lambda_i)+R\bar{g}_{i+1}(\Lambda_i)] \tag{A3}$$

$$\bar{g}_{M-1}'(\Lambda_i) = (R+\nabla^2 J_M(x_0,\Lambda_i))^{-1}\nabla^2 J_M(x_0,\Lambda_i) - (R+\nabla^2 J_M(x_0,\Lambda_i))^{-2}\times J_M'''(x_0,\Lambda_i)\nabla J_M(x_0,\Lambda_i) \tag{A4}$$

At the optimal point Λ_*, where $\nabla J_M(x_0,\Lambda_*)=0$ and $\bar{g}_i(\Lambda_*)=0$, Eqs. (A3) and (A4) reduce to

$$\bar{g}_i'(\Lambda_*) = (R+\nabla^2 J_M(x_0,\Lambda_*))^{-1}[\nabla^2 J_M(x_0,\Lambda_*)+ R\bar{g}_{i+1}'(\Lambda_*)] \tag{A5}$$

$$\bar{g}_{M-1}'(\Lambda_*) = (R+\nabla^2 J_M(x_0,\Lambda_*))^{-1}\times\nabla^2 J_M(x_0,\Lambda_*) \tag{A6}$$

By backward induction, it follows that

$$\bar{g}_i'(\Lambda_*) = I - [(R+\nabla^2 J_M(x_0,\Lambda_*))^{-1}R]^{M-i} \tag{A7}$$

Consequently, for the update rule in Eq. (11), one has

$$\begin{aligned}\Lambda_{i+1}-\Lambda_* &= \Lambda_i - \bar{g}_i - \Lambda_* = \Lambda_i - \Lambda_* - \\ &[\bar{g}_i(\Lambda_*) + \bar{g}_i'(\Lambda_i-\Lambda_*) + o(|\Lambda_i-\Lambda_*|)] \approx \\ &\Lambda_i-\Lambda_* - \bar{g}_i'(\Lambda_i-\Lambda_*) = \\ &(I-\bar{g}_i')(\Lambda_i-\Lambda_*) = \\ &[(R+\nabla^2 J_M(x_0,\Lambda_i))^{-1}R]^{M-i}(\Lambda_i-\Lambda_*).\end{aligned}$$

Since $\nabla^2 J_M(x_0,\Lambda_i) > 0$, it follows that $\|(R+\nabla^2 J_M(x_0,\Lambda_i))^{-1} R\| < 1$, which establishes the contraction property required in Lemma 3. ■

APPENDIX D
PROOF OF THEOREM 1

Proof Since $\nabla^2 J_M(x_0,\Lambda_*) > 0$ and $R > 0$, the matrix $R+\nabla^2 J_M(x_0,\Lambda_*)$ is positive definite and hence

$$\|(R+\nabla^2 J_M(x_0,\Lambda_*))^{-1}R\| < 1.$$

From the update rule in Eq. (11), we obtain

$$|\Lambda_{i+1}-\Lambda_*| = |\Lambda_i - \bar{g}_i(\Lambda_i) - \Lambda_*| = |\Lambda_i - \Lambda_* - \bar{g}_i(\Lambda_i)| \tag{A8}$$

A. Second-Order Expansion of the Update Map

Applying the second-order Taylor expansion of $\bar{g}_i(\Lambda_i)$ around Λ_* yields

$$\bar{g}_i(\Lambda_i) = \bar{g}_i(\Lambda_*) + \bar{g}_i'(\Lambda_*)(\Lambda_i-\Lambda_*) + \bar{g}_i''(\Lambda_*)(\Lambda_i-\Lambda_*)^2 + o(|\Lambda_i-\Lambda_*|^2).$$

Using $\bar{g}_i(\Lambda_*)=0$ and substituting into Eq. (A8), we obtain

$$|\Lambda_{i+1}-\Lambda_*| \leqslant \|(I-\bar{g}_i'(\Lambda_*))\||\Lambda_i-\Lambda_*| + \|\bar{g}_i''(\Lambda_*)\||\Lambda_i-\Lambda_*|^2 + o(|\Lambda_i-\Lambda_*|^2) \tag{A9}$$

By Lemma 3, the first-order term satisfies

$$\|(I-\bar{g}_i'(\Lambda_*))\| = \|(R+\nabla^2 J_M(x_0,\Lambda_*))^{-1}R\|^{M-i},$$

which establishes linear contraction.

B. Computation of the Second Derivative $\bar{g}_i''$

To characterize the higher-order term, we differentiate Eqs. (A5) and (A6) with respect to Λ_i, which yields

$$\begin{aligned}\bar{g}_i''(\Lambda_i) &= (R+\nabla^2 J_M(x_0,\Lambda_i))^{-1}[J_M'''(x_0,\Lambda_i)+ \\ &R\bar{g}_{i+1}''(\Lambda_i)] - (R+\nabla^2 J_M(x_0,\Lambda_i))^{-2}\times \\ &J_M'''(x_0,\Lambda_i)[\nabla^2 J_M(x_0,\Lambda_i)+R\bar{g}_{i+1}'(\Lambda_i)] - \\ &(R+\nabla^2 J_M(x_0,\Lambda_i))^{-2}[J_M^{(4)}(x_0,\Lambda_i)(\nabla J_M + \\ &R\bar{g}_{i+1}) + J_M'''(x_0,\Lambda_i)(\nabla^2 J_M + R\bar{g}_{i+1}')] + \\ &(R+\nabla^2 J_M(x_0,\Lambda_i))^{-3}(J_M'''(x_0,\Lambda_i))^2\times \\ &(\nabla J_M + R\bar{g}_{i+1})\end{aligned} \tag{A10}$$

For the terminal index $i=M-1$, differentiation gives

$$\begin{aligned}\bar{g}_{M-1}''(\Lambda_i) &= (R+\nabla^2 J_M(x_0,\Lambda_i))^{-1}J_M'''(x_0,\Lambda_i) - \\ &(R+\nabla^2 J_M(x_0,\Lambda_i))^{-2}[\nabla^2 J_M J_M''' + \\ &J_M^{(4)}\nabla J_M] + 2(R+ \\ &\nabla^2 J_M(x_0,\Lambda_i))^{-3}(J_M''')^2\nabla J_M.\end{aligned}$$

C. Evaluation at the Optimal Point

At $\Lambda_i=\Lambda_*$, we have $\nabla J_M(x_0,\Lambda_*)=0$, which simplifies Eq. (A10) to

$$\begin{aligned}\bar{g}_i''(\Lambda_*) &= (R+\nabla^2 J_M(x_0,\Lambda_*))^{-1}\times \\ &[J_M'''(x_0,\Lambda_*)+R\bar{g}_{i+1}''(\Lambda_*)] - \\ &2(R+\nabla^2 J_M(x_0,\Lambda_*))^{-2}J_M'''(x_0,\Lambda_*)\times \\ &[\nabla^2 J_M(x_0,\Lambda_*)+R\bar{g}_{i+1}'(\Lambda_*)].\end{aligned}$$

By backward induction, this recursion yields the closed-form expression

$$\begin{aligned}\bar{g}_i''(\Lambda_*) &= (\nabla^2 J_M(x_0,\Lambda_*))^{-1}J_M'''(x_0,\Lambda_*)\times \\ &(R+\nabla^2 J_M(x_0,\Lambda_*))^{-(M-i+1)}\times \\ &[(2M-2i+1)R^{M-i}\nabla^2 J_M(x_0,\Lambda_*)+ \\ &R^{M-i+1} - (R+\nabla^2 J_M(x_0,\Lambda_*))^{M-i+1}].\end{aligned}$$

Since

$$|(2M-2i+1)R^{M-i}\nabla^2 J_M + R^{M-i+1}| < |(R+\nabla^2 J_M)^{M-i+1}|,$$

it follows that

$$\|\bar{g}_i''(\Lambda_*)\| < \|(\nabla^2 J_M(x_0,\Lambda_*))^{-1}J_M'''(x_0,\Lambda_*)\|.$$

D. Superlinear Convergence

Substituting the above bound into Formula (A9) establishes

$$|\Lambda_{i+1}-\Lambda_*| \leqslant c|\Lambda_i-\Lambda_*| + O(|\Lambda_i-\Lambda_*|^2).$$

Moreover, since $c=\|(R+\nabla^2 J_M(\Lambda_*))^{-1}R\|^{M-i} \to 0$ as $M\to\infty$ for any fixed i, the quadratic term becomes dominant, which yields local superlinear convergence. This completes the proof. ■

APPENDIX E
PROOF OF ALGORITHM 2

This section provides a technical justification for the recursive structure of Algorithm 2. The proof is established via backward induction on the iteration index i, together with first-order local linearization of the gradient $\nabla J_M(x_0,\Lambda)$.

Proof

Step 1: Base case ($i = M-1$)

For $i = M-1$, the update equation reads

$$\Lambda_M = \Lambda_{M-1} + \tau_{M-1}.$$

Applying the definition of τ_{M-1} and a first-order Taylor expansion of $\nabla J_M(x_0, \Lambda_M)$ around Λ_{M-1} yields

$$\Lambda_M \approx \Lambda_{M-1} - R^{-1}[\nabla J_M(x_0, \Lambda_{M-1}) + \nabla^2 J_M(x_0, \Lambda_{M-1})(\Lambda_M - \Lambda_{M-1})].$$

Rearranging terms gives

$$\Lambda_M - \Lambda_{M-1} = -\varXi \nabla J_M(x_0, \Lambda_{M-1}),$$

where $\varXi = (R + \nabla^2 J_M(x_0, \Lambda_{M-1}))^{-1}$. Therefore, we have

$$\Lambda_M = \Lambda_{M-1} - \hat{g}_{M-1}, \qquad \hat{g}_{M-1} = \varXi \nabla J_M(x_0, \Lambda_{M-1}).$$

Step 2: Induction hypothesis

Assume that for index $i+1 \leqslant M-1$, the update relation holds for all $k = i+1, i+2, \ldots, M-1$

$$\Lambda_{k+1} = \Lambda_k - \hat{g}_k,$$

where $\hat{g}_k$ is computed recursively.

Step 3: Induction step

Consider the update at index i. From the definition of the accumulated dual variable

$$\zeta_i = \sum_{j=i}^{M} \nabla J_M(x_0, \Lambda_j),$$

the corresponding update direction satisfies

$$\tau_i = -R^{-1} \sum_{j=i+1}^{M} \nabla J_M(x_0, \Lambda_j).$$

Substituting this expression into the update $\Lambda_{i+1} = \Lambda_i + \tau_i$ and applying first-order Taylor expansions to $\nabla J_M(x_0, \Lambda_j)$ for all $j > i$, together with the induction hypothesis $\Lambda_j - \Lambda_{j-1} = -\hat{g}_{j-1}$, yields

$$\Lambda_{i+1} \approx \Lambda_i - R^{-1}[\nabla J_M(x_0, \Lambda_i) + \sum_{j=i+1}^{M-1} (\nabla J_M(x_0, \Lambda_j) - \nabla^2 J_M(x_0, \Lambda_j)\hat{g}_j)].$$

Rearranging the above expression leads to

$$\Lambda_{i+1} = \Lambda_i - \hat{g}_i,$$

where

$$\hat{g}_i = \varXi[\nabla J_M(x_0, \Lambda_i) + \sum_{j=i+1}^{M-1} (\nabla J_M(x_0, \Lambda_j) - \nabla^2 J_M(x_0, \Lambda_j)\hat{g}_j)].$$

Step 4: Recursive Hessian-free form

To obtain the practical recursion used in Algorithm 2, all gradient and Hessian terms on the right-hand side are evaluated at the common reference point Λ_i. This approximation yields the recursive relation

$$\hat{g}_i(\Lambda_i) = \varXi \nabla J_M(x_0, \Lambda_i) + [I - \varXi \nabla^2 J_M(x_0, \Lambda_i)]\hat{g}_{i+1}(\Lambda_i),$$

with terminal condition

$$\hat{g}_{M-1}(\Lambda_i) = \varXi \nabla J_M(x_0, \Lambda_i).$$

This recursion coincides exactly with the update rule stated in Algorithm 2, thereby completing the proof. ■

ACKNOWLEDGMENT

This work was supported by the Support Plan on Science and Technology for Youth Innovation of Universities in Shandong Province (No. 2021KJ086) and the National Natural Science Foundation of China (No. 62003234).

REFERENCES

[1] K. L. Teo, B. Li, C. Yu, and V. Rehbock, *Applied and Computational Optimal Control*: *A Control Parametrization Approach*. Cham, Germany: Springer, 2021.

[2] X. Zhu, C. Yu, and K. L. Teo, Sequential adaptive switching time optimization technique for optimal control problems, *Automatica*, 2022, 146, 110565.

[3] T. Sun, K. L. Teo, and X. Sun, Numerical optimal control for switched nonlinear systems with inequality path constraints, *Syst. Control Lett.*, 2023, 182, 105653.

[4] A. D. Belegundu and T. R. Chandrupatla, *Optimization Concepts and Applications in Engineering*. Cambridge, UK: Cambridge University Press, 2019.

[5] E. K. P. Chong, W.-S. Lu, and S. H. Żak, *An Introduction to Optimization*: *With Applications to Machine Learning*. Hoboken, NJ, USA: John Wiley & Sons, 2024.

[6] B. Li, Q. Li, Y. Zeng, Y. Rong, and R. Zhang, 3D trajectory optimization for energy-efficient UAV communication: A control design perspective, *IEEE Trans. Wireless Commun.*, 2022, 21(6), 4579–4593.

[7] W. Bai, L. Cao, G. Dong, and H. Li, Adaptive reinforcement learning tracking control for second-order multi-agent systems, in *Proc. 2019 IEEE 8th Data Driven Control and Learning Systems Conference*, Dali, China, 2019, 202–207.

[8] L. Zhang, H. Zhang, and Z. Wang, Analytical solution to optimal distributed bipartite consensus for heterogeneous multiagent systems on coopetition networks: A fast convergent algorithm, *IEEE Trans. Autom. Control*, 2026, 71(4), 2762–2769.

[9] J.-X. Zhang, W.-L. Qi, and T. Chai, Constrained tracking control of MIMO nonlinear systems with discontinuous references and unknown dynamics, *IEEE Trans. Autom. Sci. Eng.*, 2025, 22, 15139–15149.

[10] Q. Liu, L. Zhang, and H. Zhang, Distributed multi-objective optimal control for DC distribution networks based on LQR, *IEEE Trans. Power Syst.*, 2026, 41(3), 1675–1687.

[11] A. S. Bedi, A. Koppel, and K. Rajawat, Asynchronous online learning in multi-agent systems with proximity constraints, *IEEE Trans. Signal Inf. Process. Networks*, 2019, 5(3), 479–494.

[12] D. Lee, N. He, P. Kamalaruban, and V. Cevher, Optimization for reinforcement learning: From a single agent to cooperative agents, *IEEE Signal Process. Mag.*, 2020, 37(3), 123–135.

[13] Z. Dong and Z. You, Optimal guaranteed cost control of a class of nonlinear uncertain discrete-time systems: An LMI approach, in *Proc. 2006 American Control Conference*, Minneapolis, MN, USA, 2006, 4.

[14] C. Lv, H. Li, and H. Zhang, Finite-horizon discrete-time optimal control for nonlinear systems under state and control constraints, arXiv preprint arXiv: 2503.15794, 2025.

[15] Z. Guo, Y. Sun, Y. Xu, L. Zhang, and H. Zhang, Distributed optimization method based on optimal control, arXiv preprint arXiv: 2411.10658, 2025.

[16] T. Docekal and S. Ozana, Application of two-point boundary problem with optimization of a candidate function, in *Proc. 2020 20th International Conference on Control*, *Automation and Systems*, Busan, Republic of Korea, 2020, 912–915.

[17] H. Wang, Y. Xu, Z. Guo, and H. Zhang, Superlinear optimization algorithms, arXiv preprint arXiv: 2403.11115, 2025.

[18] C. Lv, X. Yin, H. Li, and H. Zhang, Optimal control of nonlinear systems using accelerated gradient and Hessian computation, *IEEE Trans. Ind. Electron.*, 2026, 73(3), 4645–4653.

[19] H. Zhang, H. Wang, Y. Xu, and Z. Guo, Optimization methods rooted in optimal control, *Sci. China Inf. Sci.*, 2024, 67(12), 222208.

[20] S. Y. Lim and K. H. Lim, Second-order derivative optimization methods in deep learning neural networks, in *Proc. 2022 International*

Conference on Green Energy, *Computing and Sustainable Technology*, Miri Sarawak, Malaysia, 2022, 470–475.

[21] O. Shamir, N. Srebro, and T. Zhang, Communication-efficient distributed optimization using an approximate Newton-type method, in *Proc. 31st International Conference on Machine Learning*, Beijing, China, 2014, II-1000–II-1008.

[22] D. Bajović, D. Jakovetić, N. Krejić, and N. K. Jerinkić, Newton-like method with diagonal correction for distributed optimization, *SIAM J. Optim.*, 2017, 27(2), 1171–1203.

[23] S. Soori, K. Mishchenko, A. Mokhtari, M. M. Dehnavi, and M. Gurbuzbalaban, DAve-QN: A distributed averaged quasi-newton method with local superlinear convergence rate, in *Proc. 23rd International Conference on Artificial Intelligence and Statistics*, Palermo, Italy, 2020, 1965–1976.

[24] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, Inexact Newton methods, *SIAM J. Numer. Anal.*, 1982, 19(2), 400–408.

[25] M. Eisen, A. Mokhtari, and A. Ribeiro, A primal-dual quasi-newton method for exact consensus optimization, *IEEE Trans. Signal Process.*, 2019, 67(23), 5983–5997.

[26] H. Wei, Z. Qu, X. Wu, H. Wang, and J. Lu, Decentralized approximate Newton methods for convex optimization on networked systems, *IEEE Trans. Control Network Syst.*, 2021, 8(3), 1489–1500.

[27] O. Shorinwa and M. Schwager, Distributed quasi-Newton method for multi-agent optimization, *IEEE Trans. Signal Process.*, 2024, 72, 3535–3546.

[28] J. Zhang, K. You, and T. Başar, Distributed adaptive Newton methods with global superlinear convergence, *Automatica*, 2022, 138, 110156.

[29] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2009.

[30] N. Qian, On the momentum term in gradient descent learning algorithms, *Neural Networks*, 1999, 12(1), 145–151.

[31] L. Bottou, Large-scale machine learning with stochastic gradient descent, in *Proc. 19th International Conference on Computational Statistics*, Paris, France, 2010, 177–186.

[32] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai, Gradient descent finds global minima of deep neural networks, in *Proc. 36th International Conference on Machine Learning*, Long Beach, CA, USA, 2019, 1675–1685.

[33] Z. Fei, Z. Wu, Y. Xiao, J. Ma, and W. He, A new short-arc fitting method with high precision using Adam optimization algorithm, *Optik*, 2020, 212, 164788.

[34] B. O'Donoghue and E. Candès, Adaptive restart for accelerated gradient schemes, *Found. Comput. Math.*, 2015, 15, 715–732.

[35] J. Duchi, E. Hazan, and Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, *J. Mach. Learn. Res.*, 2011, 12, 2021–2159.

[36] H. Ni and B. Sun, An adaptive optimization approach for Nash equilibrium in multi-AUV confrontation games, in *Proc. 2025 7th International Conference on Robotics*, *Intelligent Control and Artificial Intelligence*, Hangzhou, China, 2025, 227–230.

[37] Q. Meng and Q. Ma, Distributed Nash equilibrium seeking for games in uncertain nonlinear systems via adaptive backstepping approach, *IEEE Trans. Control Network Syst.*, 2025, 12(1), 1188–1198.

[38] S. Chen, Y. Wan, and J. Cao, Adaptive Nash equilibrium seeking in hybrid heterogeneous open multi-agent systems under DoS attacks, *IEEE Trans. Network Sci. Eng.*, 2025, 12(4), 2770–2782.

[39] J.-Z. Xu, Z.-W. Liu, D.-X. He, Z. Jia, and M.-F. Ge, Dynamic Nash equilibrium seeking for constrained noncooperative game of open multiagent systems, *IEEE Trans. Syst., Man, Cybern.: Syst.*, 2025, 55(6), 3846–3855.

[40] C. Li, A. Li, Y. Wu, and L. Wang, Logical dynamic games: Models, equilibria, and potentials, *IEEE Trans. Autom. Control*, 2024, 69(11), 7584–7599.

[41] Y. Huang, J. Sun, and Q. Fei, Distributed event-triggered Nash equilibrium seeking for aggregative game with second-order dynamics, *IEEE/CAA J. Autom. Sin.*, 2025, 12(7), 1519–1521.

[42] M. Sun, J. Liu, L. Ren, and C. Sun, Fixed-time consensus-based Nash equilibrium seeking, *IEEE/CAA J. Autom. Sin.*, 2024, 11(1), 267–269.

[43] K. Lu, H. Zhang, and L. Wang, Aggregative games with bilevel structures: Distributed algorithms and convergence analysis, *IEEE Trans. Autom. Control*, to be published.

[44] X. Li, M. Bai, H. Zhang, Z. Jing, P. Ju, and Z. Guo, Incentive Stackelberg game for H_∞-constrained multi-hierarchy systems under observation information, *IET Control Theory Appl.*, 2022, 16(18), 1821–1833.

[45] Z. Jing, X. Li, P. Ju, and H. Zhang, Optimal control and filtering for hierarchical decision problems with H_∞ constraint based on Stackelberg strategy, *IEEE Trans. Autom. Control*, 2024, 69(9), 6238–6245.

[46] A. Nedic, A. Ozdaglar, and P. A. Parrilo, Constrained consensus and optimization in multi-agent networks, *IEEE Trans. Autom. Control*, 2010, 55(4), 922–938.

[47] S. Ruder, An overview of gradient descent optimization algorithms, arXiv preprint arXiv: 1609.04747, 2017.

[48] A. G. Baydin, R. Cornish, D. Martínez-Rubio, M. Schmidt, and F. Wood, Online learning rate adaptation with hypergradient descent, in *Proc. 6th International Conference on Learning Representations*, Vancouver, Canada, 2018.

Xiaoqian Li received the ME degree in control engineering from Northeastern University, Shenyang, China, in 2013, and the PhD degree in control science and engineering from Shandong University, Jinan, China, in 2019. She is currently an associate professor at Taishan University, Taian, China. Her research interests include optimal control, reinforcement learning, and game theory.

Zonglei Jing received the ME degree from Beihang University, Beijing, China, in 2024. He is currently pursuing the PhD degree at State Key Laboratory of Complex and Critical Software Environment, School of Computer Science and Engineering, Beihang University, China. His research interests include adversarial attacks, generative model safety, reinforcement learning, and AI safety testing and evaluation.

Zhongjin Guo received the MS degree in fundamental mathematics from Wuhan University, Wuhan, China, in 2006, and the PhD degree in civil and architectural engineering from City University of Hong Kong, Hong Kong, China, in 2012. From 2015 to 2017, he was a postdoctoral research fellow at Beijing University of Technology, Beijing, China. He is currently a professor at Taishan University, Taian, China. His research interests include nonlinear vibration, fractional-order dynamical systems, and related applications in applied mathematics.

Peijun Ju received the BE degree in mathematics from Qufu Normal University, Jining, China, in 2000, and the ME degree in mathematics and the PhD degree in control science and engineering from Shandong University, Jinan, China, in 2004 and 2017, respectively. He is currently a professor at Taishan University, China. His research interests include time delay systems, optimal control, and game theory.

Shen Zhang received the ME degree in engineering management from Shandong University of Science and Technology, Shandong, China, in 2017, and the PhD degree in control science and engineering from Shandong University, Jinan, China, in 2024. He is currently a lecturer at School of Mathematics and Statistics, Taishan University, Taian, Shandong, China. His research interests include optimal control, distributed control, stabilization, game theory, and stochastic systems.