Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 9(10): 27272–27292.
DOI: 10.3934/math.20241325
Received: 27 July 2024
Revised: 28 August 2024
Accepted: 04 September 2024
Published: 20 September 2024

Research article

Distributed Newton method for time-varying convex optimization with backward Euler prediction

Zhuo Sun[1], **Huaiming Zhu**[1,*] **and Haotian Xu**[2]

[1] College of Transportation Engineering, Dalian Maritime University, 116026 Dalian, China

[2] Information Science and Technology College, Dalian Maritime University, 116026 Dalian, China

* **Correspondence:** Email: zhu_huaiming@foxmail.com.

Abstract: We investigated the challenge of unconstrained distributed optimization with a time-varying objective function, employing a prediction-correction approach. Our method introduced a backward Euler prediction step that used the differential information from consecutive moments to forecast the trajectory's future direction. This predicted value was then refined through an iterative correction process. Our analysis and experimental results demonstrated that this approach effectively addresses the optimization problem without requiring the computation of the Hessian matrix's inverse.

Keywords: unconstrained optimization; prediction-correction strategy; backward Euler prediction
Mathematics Subject Classification: 49M15, 90C25

1. Introduction

Time-varying distributed optimization (TVDO) has gained increasing attention due to its practical advantages over time-invariant distributed optimization (TIDO) in dynamic environments where the objective function or constraints may evolve over time. TVDO has been applied in various fields, including power systems [1], traffic system [2], and robotics [3], with further applications in energy management [4]. In TVDO, the optimal solution changes over time, making traditional algorithms designed for TIDO, which aim to reach a static optimizer, unsuitable for direct application. To address TVDO, several discrete-time algorithms (DTAs) have been developed [5–8]. For instance, prediction-correction methods are employed in [6, 7] to solve TVDO with a specific sampling period, where the tracking error is linked to the size of the sampling period. A comprehensive review of DTAs and related work can be found in the survey [9]. However, due to factors like sampling period, step size, or errors in local optimization, continuous-time algorithms (CTAs) generally face challenges in asymptotically tracking the time-varying optimal trajectory.

Given the extensive use of digital computing and sensing units [10, 11], we focus on a discrete-time framework. Specifically, we sample the objective functions $F(x;t)$ at discrete time instances t_k, where $k = 0, 1, 2, \ldots$ and the sampling period is defined as $h \triangleq t_{k+1} - t_k$. Instead of addressing the time-varying problem directly, we solve a series of time-invariant problems. If the sampling period h is made sufficiently small, the resulting solution trajectory $F(t_k)$ can be obtained with high accuracy. However, in most practical applications, solving these problems for each time sample is impractical, as the computation time required to find each optimizer often surpasses the rate at which the solution trajectory evolves, unless $F(t)$ remains nearly stationary.

The batch method is one traditional approach for addressing the problem sequence, where the objective function is sampled at specified intervals, solving each resultant static problem within given periods. However, this method often fails to align with real-time processing demands, particularly when computational resources are limited and sampling intervals are short. This limitation becomes more pronounced with larger problem sizes, preventing convergence within the required timeframe [12]. Consequently, attention has shifted towards online optimization techniques. These methods update the optimization problem continuously throughout the algorithm's iterations, allowing for the extraction of suboptimal solutions at any stage, regardless of convergence status [1, 13]. Over time, these solutions increasingly approximate the optimal solution. In pursuit of improving this approach, various strategies have been developed. For example, a method based solely on correction operations was introduced, achieving an asymptotic error bound on the order of $O(h)$ [14]. Another strategy involves a prediction-correction algorithm tailored for time-varying parameter optimization, though it requires an initial optimal solution, limiting its practical application [15]. Nonetheless, these methods have facilitated some theoretical advances in reducing the computational load, especially in convex optimization scenarios. Further, the interior point method has been applied to solve constrained convex optimization problems characterized by time-varying elements, utilizing a log-barrier penalty function and a dynamic system comprising predictive and corrective components based on Newton's method [16]. For unconstrained time-varying optimization, a suite of algorithms deploying prediction-correction strategies has been proposed. These include Gradient Trajectory Tracking (GTT) and Approximate Gradient Tracking (AGT), which achieve an $O(h^2)$ asymptotic error range. Moreover, Newton Trajectory Tracking (NTT) and Approximate Newton Tracking (ANT) are shown to offer superior error bounds [10]. Building on this framework, a decentralized prediction-correction algorithm for time-varying network optimization challenges has been developed, employing novel matrix splitting techniques and approximating matrix inverses using the Taylor series [6, 17].

Prediction-correction algorithms [18], utilizing nonstationary optimization techniques [15,19,20], have been developed to iteratively solve convex programs that change over time. These algorithms function by predicting, at time t_k, the optimal solution for the next time step t_{k+1}, using an approximation of how the objective function F varies during this interval. The initial prediction is then refined through gradient or Newton descent methods. However, these algorithms are tailored for centralized systems. In our work, we address time-varying convex programs in decentralized environments, where nodes can communicate only with their immediate neighbors. Therefore, the prediction-correction methods proposed in [18] are not directly applicable in this decentralized context.

In the discussed algorithms that incorporate a prediction step, while achieving tighter error bounds, these methods necessitate the computation or approximation of the inverse of the Hessian matrix. In certain scenarios, this process also involves calculating mixed partial derivatives. It is established that computing the inverse of a matrix is computationally intensive, with the complexity typically represented as $O(p^3)$, where p is the matrix's dimension. This complexity escalates further for objective functions that are more intricate due to the requirement of mixed partial derivative calculations. In this paper, we outline our major contributions as follows:

1) We introduce a backward Euler prediction step tailored for the unconstrained time-varying optimization problem. This approach eliminates the need for calculating matrix inverses and computing mixed partial derivatives, resulting in reduced computational complexity. The corrective aspect of the algorithm is derived from a Newton step.

2) We establish the convergence of our proposed algorithm and detail its convergence rate. The asymptotic error associated with our algorithm depends on the sampling period h, ranging from a worst-case scenario of $O(h^2)$ to an optimal scenario of $O(h^4)$.

The structure of this paper is as follows: In Section II, we provide the mathematical foundations essential for the development of the major results. In Section III, we detail the algorithms incorporating the backward Euler prediction step. The performance of these algorithms is examined in Section IV. A numerical example demonstrating the practical application of our theoretical findings is presented in Section V. We conclude in Section VI.

2. Preliminaries

We focus on a connected, undirected graph $\mathcal{G} = (V, E)$, where the vertex set V consists of n nodes, and the edge set E consists of m edges. Distributed optimization algorithms are employed to address the problem of minimizing a global smooth, strongly convex cost function across a set of nodes, where the objective function is expressed as a sum of local functions. We intend to solve the following problem raised in [6]

$$\mathbf{x}^*(t): \; = \underset{\mathbf{x}\in\mathbb{R}^{np}}{\arg\min}\, F(\mathbf{x};t): \; = f(\mathbf{x};t) + g(\mathbf{x};t), \;\; t \geq 0, \tag{2.1}$$

where $\mathbf{x} \in \mathbb{R}^{np}$ represents the stacked vector of decision variables for all nodes, and $\mathbf{x}^i \in \mathbb{R}^p$ denotes the decision variable for node i, and $\mathbf{x} = (\mathbf{x}^{1^T}; \ldots; \mathbf{x}^{n^T})^T$. The function $f(\mathbf{x};t): \; = \sum\limits_{i\in V} f^i(\mathbf{x}^i;t)$ is the sum of the locally available functions at each node, while $g(\mathbf{x};t): \; = \sum\limits_{i\in V} g^{i,i}(\mathbf{x}^i;t) + \sum\limits_{(i,j)\in E} g^{i,j}(\mathbf{x}^i, \mathbf{x}^j;t)$ incorporates terms induced by the network structure $\mathcal{G}$, necessitating coordination and information exchange across the network. Before introducing distributed protocols to solve the problem in (2.1), we present an example to clarify the problem setting.

For example, in a wireless sensor network, each sensor node must adhere to channel capacity and interference constraints, which can be formulated as a resource allocation problem. The time-varying utility function for sensor i is defined as $f^i : \mathbb{R}^p \times \mathbb{R}_+ \to \mathbb{R}$, with the decision variable $\mathbf{x}^i \in \mathbb{R}^p$ representing the resources allocated to node i in a network $\mathcal{G}$ consisting of n sensors. This leads to the formulation of the time-varying resource allocation problem.

$$\min_{\mathbf{x}^i\in\mathbb{R}^p,\ldots\mathbf{x}^n\in\mathbb{R}^p} \sum_{i\in V} f^i(\mathbf{x}^i;t) \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b}(t), \tag{2.2}$$

where the matrix $\mathbf{A} \in \mathbb{R}^{lp\times np}$ is the augmented graph edge incidence matrix. This matrix is composed of $l \times n$ square blocks, each of dimension p. For an edge $e = (j, k)$, where $j < k$, that connects node j

to node k, the block (e, j) in $\mathbf{A}$ is given by $[\mathbf{A}]_{ej} = \mathbf{I}_p$, and the block $[\mathbf{A}]_{ek} = -\mathbf{I}_p$, where $\mathbf{I}_p$ denotes the identity matrix of dimension p. All other blocks in $\mathbf{A}$ are zeros. The time-varying vectors $\mathbf{b}(t) \in \mathbb{R}^{lp}$ are determined by channel capacity and rate transmission constraints.

In our work, we consider the nodes to be consumer devices aiming to solve decentralized approximations of the problem defined in (2.2). Specifically, we explore the approximate augmented Lagrangian relaxation of (2.2), and solve

$$\min_{\mathbf{x}^i \in \mathbb{R}^p, \ldots \mathbf{x}^n \in \mathbb{R}^p} \sum_{i \in V} f^i(\mathbf{x}^i; t) + \frac{1}{\beta^2} \|\mathbf{A}\mathbf{x} - \mathbf{b}(t)\|^2, \tag{2.3}$$

where the parameter $\beta > 0$ acts as a regularization term, encouraging consistency among all nodes. In (2.3), the matrix $\mathbf{A} \in \mathbb{R}^{lp \times np}$ is the graph edge incidence matrix. It is straightforward to observe that the first term in (2.3) is identical to the first term in (2.1). Moreover, given the definition of the augmented graph edge incidence matrix $\mathbf{A}$, we can simplify the second term in (2.3) to $\left\|(\mathbf{x}_i - \mathbf{x}_j) - \mathbf{b}_i(t)\right\|^2$, which corresponds to the functions $g^{i,j}(\mathbf{x}_i, \mathbf{x}_j; t)$ in (2.1).

Notation: Let $\|\cdot\|$ represent the Euclidean norm. The gradient of the function $F(\mathbf{x}; t)$ with respect to $\mathbf{x}$ is denoted by $\nabla_{\mathbf{x}} F(\mathbf{x}; t)$. The partial derivatives of $\nabla_{\mathbf{x}} F(\mathbf{x}; t)$ with respect to $\mathbf{x}$ and t are indicated by $\nabla_{\mathbf{xx}} F(\mathbf{x}; t)$ and $\nabla_{t\mathbf{x}} F(\mathbf{x}; t)$, respectively. The third-order derivative of $F(\mathbf{x}; t)$ with respect to $\mathbf{x}$ is denoted as $(\nabla_{\mathbf{xxx}} F(\mathbf{x}; t)$. The time derivative of the Hessian of $F(\mathbf{x}; t)$ with respect to t is expressed as $\nabla_{t\mathbf{xx}} F(\mathbf{x}; t) = \nabla_{\mathbf{x}t\mathbf{x}} F(\mathbf{x}; t)$. Finally, the second derivative of $\nabla_{\mathbf{x}} F(\mathbf{x}; t)$ with respect to t is represented by $\nabla_{tt\mathbf{x}} F(\mathbf{x}; t)$.

3. Algorithm

3.1. The dynamical system

If $\mathbf{x}^*$ represents the optimal solution for the objective function $F(\mathbf{x}; t)$, then the gradient of $F(\mathbf{x}; t)$ with respect to $\mathbf{x}$ must satisfy

$$\nabla_{\mathbf{x}} F(\mathbf{x}^*; t) \equiv 0, \quad \forall t \geq 0. \tag{3.1}$$

As a consequence, the time derivative of this expression must also be zero, leading to

$$\begin{aligned} \frac{d\nabla_{\mathbf{x}} F(\mathbf{x}^*; t)}{dt} &= \nabla_{\mathbf{xx}} F(\mathbf{x}^*; t)\dot{\mathbf{x}}^*(t) + \nabla_{t\mathbf{x}} F(\mathbf{x}^*; t) \\ &= 0, \end{aligned} \tag{3.2}$$

where $\dot{\mathbf{x}}^*(t)$ represents the time derivative of $\mathbf{x}^*(t)$. If the Hessian of $F(\mathbf{x}; t)$ is invertible, we can deduce from the above equation that

$$\dot{\mathbf{x}}^*(t) = -\nabla_{\mathbf{xx}} F(\mathbf{x}^*; t)^{-1} \nabla_{t\mathbf{x}} F(\mathbf{x}^*; t). \tag{3.3}$$

We can apply either the gradient descent method or the Newton method to optimize $F(\mathbf{x}; t)$, leading to the formulation of a continuous-time dynamical system as follows:

$$\dot{\mathbf{x}}(t) = -\gamma \nabla_{\mathbf{xx}} F(\mathbf{x}; t)^{-1} \nabla_{\mathbf{x}} F(\mathbf{x}; t), \tag{3.4}$$

where γ represents a control gain with the constraint $0 < \gamma \leq 1$. When $\gamma = \alpha \nabla_{\mathbf{xx}} F(\mathbf{x}; t)$, where $\alpha > 0$ is a constant, this corresponds to the gradient term in Eq (8). Conversely, when $\gamma = 1$, Eq (8) represents the Newton term. The trajectory $\mathbf{x}(t)$ generated by Eq (8) will approach a vicinity of $\mathbf{x}^*(t)$, but due to the time-varying nature of the solution, it does not converge precisely to $\mathbf{x}^*(t)$ [21].

If the optimal solution $\mathbf{x}^*(t)$ was known for some $t_0 \geq 0$, the system (3.3) could be used to track the evolution of $\mathbf{x}^*(t)$, since (3.3) guarantees the optimality condition $\nabla_{\mathbf{x}} F(\mathbf{x}^*; t) = 0$ for all $t \geq t_0$. If $\mathbf{x}^*(t)$ is not accessible at any time, we can combine the dynamics (3.3) and (3.4) and devise the following dynamical system:

$$\dot{\mathbf{x}}(t) = \mathcal{F}(\mathbf{x}; t), \tag{3.5}$$

where $\mathcal{F}(\mathbf{x}; t)$ is defined as

$$\mathcal{F}(\mathbf{x}; t) = -\nabla_{\mathbf{xx}} F(\mathbf{x}; t)^{-1} \nabla_{t\mathbf{x}} F(\mathbf{x}; t) - \gamma \nabla_{\mathbf{xx}} F(\mathbf{x}; t)^{-1} \nabla_{\mathbf{x}} F(\mathbf{x}; t). \tag{3.6}$$

The dynamics described in (3.6) consist of two components: A prediction component and a correction component. The prediction component, given by $-\nabla_{\mathbf{xx}} F(\mathbf{x}; t) \nabla_{t\mathbf{x}} F(\mathbf{x}; t)$, aims to forecast the change in the optimal solution (refer to (3.3)). The correction component, $-\gamma \nabla_{\mathbf{xx}} F^{-1}(\mathbf{x}; t) \nabla_{\mathbf{x}} F(\mathbf{x}; t)$, serves to steer $\mathbf{x}(t)$ toward the optimum.

3.2. Discretization calculation

For any time interval $[t_0, t_0 + T]$, where $0 < T < +\infty$, we can partition the interval into N subintervals such that $t_{k+1} = t_k + h$, for $k = 0, 1, 2, \ldots, N - 1$, where h is the discretization step size. Let $\mathbf{x}(t_k)$ represent the solution of $F(\mathbf{x}; t)$ at time t_k. For simplicity, we may denote $\mathbf{x}(t_k)$ as $\mathbf{x}_k$ throughout this paper.

A numerical method for approximating (3.5) is considered a one-step method if, for all $k \geq 0$, $\mathbf{x}_{k+1}$ depends solely on $\mathbf{x}_k$. Two examples of such one-step methods are presented in [22].

(1) The forward Euler calculation method

$$\mathbf{x}_{k+1} \approx \mathbf{x}_k + hF(\mathbf{x}_k, t_k). \tag{3.7}$$

(2) The backward Euler calculation method

$$\mathbf{x}_k \approx \mathbf{x}_{k+1} - hF(\mathbf{x}_{k+1}, t_{k+1}). \tag{3.8}$$

The primary distinction between the forward and backward methods is found in their treatment of the first-order approximation term for $F(\mathbf{x}; t)$. It follows from (3.8) that

$$\mathbf{x}_{k-1} \approx \mathbf{x}_k - hF(\mathbf{x}_k, t_k), \tag{3.9}$$

yielding

$$F(\mathbf{x}_k, t_k) \approx \frac{\mathbf{x}_k - \mathbf{x}_{k-1}}{h}. \tag{3.10}$$

Substituting (3.10) into (3.7) gives

$$\mathbf{x}_{k+1} \approx \mathbf{x}_k + h\frac{\mathbf{x}_k - \mathbf{x}_{k-1}}{h} = 2\mathbf{x}_k - \mathbf{x}_{k-1}. \tag{3.11}$$

According to (3.11), we formulate the prediction step as follows:

$$\mathbf{x}_{k+1|k} = 2\mathbf{x}_k - \mathbf{x}_{k-1}. \tag{3.12}$$

The prediction step can be computed only depend on the state information at time k and $k-1$. Compared with the prediction step in [6, 17], the prediction step in our algorithm can reduce the computation complexity significantly.

To address the time-varying optimization problem in (2.1), we sample the continuous time-varying objective function $F(\mathbf{x};t)$ at discrete times t_k, where $k = 0, 1, 2, \ldots$. This transformation turns the continuous time-varying optimization problem into a sequence of time-invariant optimization problems.

$$\mathbf{x}^*(t_k): = \arg\min_{\mathbf{x}\in\mathbb{R}^{np}} F(\mathbf{x};t_k), \quad k \geq 0. \tag{3.13}$$

In network, the prediction step can be written as (3.12). For node i, the predicted variable $\mathbf{x}^i_{k+1|k}$ is just upon the information of the node itself which given by

$$\mathbf{x}^i_{k+1|k} = 2\mathbf{x}^i_k - \mathbf{x}^i_{k-1}, \tag{3.14}$$

the $\mathbf{x}^i_{k+1|k}$ is computed through local operations, but for the correction step in our algorithm, the Newton term involving the calculate of the inverse of the matrix, which requires the global communication. To develop a decentralized protocol to correct the prediction variable $\mathbf{x}_{k+1|k}$ [18], here we apply the decentralized Newton method proposed by [6]. The decentralized Newton method is formulated as follow

$$\mathbf{x}_{k+1} = \mathbf{x}_{k+1|k} - \gamma\mathbf{c}_{k+1,(K)}, \tag{3.15}$$

where $\mathbf{c}_{k+1,(K)} \in \mathbb{R}^{np}$ is called correction direction, its definition is

$$\mathbf{c}_{k+1,(K)} = \mathbf{D}^{-\frac{1}{2}}_{k+1|k} \sum_{\tau=0}^{K} \left(\mathbf{D}^{-\frac{1}{2}}_{k+1|k}\mathbf{B}_{k+1|k}\mathbf{D}^{-\frac{1}{2}}_{k+1|k}\right)^{\tau} \mathbf{D}^{-\frac{1}{2}}_{k+1|k} \times \nabla_{\mathbf{x}}F(\mathbf{x}_{k+1|k};t_{k+1}). \tag{3.16}$$

For $\mathbf{D}^{-\frac{1}{2}}_{k+1|k} \sum_{\tau=0}^{K} \left(\mathbf{D}^{-\frac{1}{2}}_{k+1|k}\mathbf{B}_{k+1|k}\mathbf{D}^{-\frac{1}{2}}_{k+1|k}\right)^{\tau} \mathbf{D}^{-\frac{1}{2}}_{k+1|k}$, it is the K-th order approximate of $\nabla_{\mathbf{xx}}F(\mathbf{x}_{k+1|k};t_{k+1})^{-1}$, which is derived from truncations of the series [6, 21]

$$\nabla_{\mathbf{xx}}F(\mathbf{x}_{k+1|k};t_{k+1})^{-1} = \mathbf{D}^{-\frac{1}{2}}_{k+1|k} \sum_{\tau=0}^{\infty} \left(\mathbf{D}^{-\frac{1}{2}}_{k+1|k}\mathbf{B}_{k+1|k}\mathbf{D}^{-\frac{1}{2}}_{k+1|k}\right)^{\tau} \mathbf{D}^{-\frac{1}{2}}_{k+1|k}, \tag{3.17}$$

where matrices $\mathbf{D}_{k+1|k}$ and $\mathbf{B}_{k+1|k}$ are defined as

$$\begin{aligned} \mathbf{D}_{k+1|k} &:= \nabla_{\mathbf{xx}}f(\mathbf{x}_{k+1|k};t_{k+1}) + diag[\nabla_{\mathbf{xx}}g(\mathbf{x}_{k+1|k};t_{k+1})], \\ \mathbf{B}_{k+1|k} &:= diag[\nabla_{\mathbf{xx}}g(\mathbf{x}_{k+1|k};t_{k+1})] - \nabla_{\mathbf{xx}}g(\mathbf{x}_{k+1|k};t_{k+1}), \end{aligned} \tag{3.18}$$

where the $diag[\nabla_{\mathbf{xx}}g(\mathbf{x}_{k+1|k}; t_{k+1})]$ denotes the block diagonal matrix, which contains the diagonal blocks of the matrix $\nabla_{\mathbf{xx}}g(\mathbf{x}_{k+1|k}; t_{k+1})$. The matrix

$$\mathbf{D}^{ii}_{k+1|k} = \nabla_{\mathbf{x}^i\mathbf{x}^i}f^i(\mathbf{x}^i_{k+1|k}; t_{k+1}) + \nabla_{\mathbf{x}^i\mathbf{x}^i}g^{i,i}(\mathbf{x}^i_{k+1|k}; t_{k+1}) + \sum_{j\in N^i}\nabla_{\mathbf{x}^i\mathbf{x}^i}g^{i,j}(\mathbf{x}^i_{k+1|k}, \mathbf{x}^j_{k+1|k}; t_{k+1}) \tag{3.19}$$

is computed at node i. The second term in (3.19) links the decisions of node i with those of its neighboring nodes $j \in \mathcal{N}^i$. The structure of the matrix $\mathbf{B}_{k+1|k}$ is determined by the graph topology, with null diagonal blocks $\mathbf{B}^{i,i}_{k+1|k}$ and non-zero off-diagonal blocks $\mathbf{B}^{i,j}_{k+1|k}$ given by $-\nabla_{\mathbf{x}^i\mathbf{x}^j}g^{i,j}(\mathbf{x}^i, \mathbf{x}^j; t)$ whenever nodes i and j are connected. The calculation of $\mathbf{B}^{ij}_{k+1|k}$ is as follows:

$$\mathbf{B}^{ij}_{k+1|k} = -\nabla_{\mathbf{x}^i\mathbf{x}^j}g^{i,j}(\mathbf{x}^i_{k+1|k}, \mathbf{x}^j_{k+1|k}; t_{k+1}). \tag{3.20}$$

Lemma 1. *According to [21], as per the definition of the correction direction in* (3.16) *the sequence of correction directions satisfies*

$$\mathbf{c}_{k+1,(\tau+1)} = \mathbf{D}^{-1}_{k+1|k}\left(\mathbf{B}_{k+1|k}\mathbf{c}_{k+1,(\tau)} + \nabla_{\mathbf{x}}F(\mathbf{x}_{k+1|k}; t_{k+1})\right), \tag{3.21}$$

define $\mathbf{c}^i_{k+1,(K)}$ *and* $\nabla_{\mathbf{x}}F^i(\mathbf{x}_{k+1|k}; t_{k+1})$ *as the i-th component of the vector* $\mathbf{c}_{k+1,(K)}$ *and* $\nabla_{\mathbf{x}}F(\mathbf{x}_{k+1|k}; t_{k+1})$*, the recursion of* (3.21) *can be decomposed into local components*

$$\mathbf{c}^i_{k+1,(\tau+1)} = -\,(\mathbf{D}^{ii}_{k+1|k})^{-1}\left(\sum_{j\in N^i}\mathbf{B}^{ij}_{k+1|k}\mathbf{c}^j_{k+1,(\tau)} + \nabla_{\mathbf{x}}F^i(\mathbf{x}_{k+1|k}; t_{k+1})\right). \tag{3.22}$$

The gradient component in (3.22)

$$\nabla_{\mathbf{x}}F^i(\mathbf{x}_{k+1|k}; t_{k+1}) = \nabla_{\mathbf{x}^i}f^i(\mathbf{x}^i_{k+1|k}; t_{k+1}) + \nabla_{\mathbf{x}^i}g^{i,i}(\mathbf{x}^i_{k+1|k}; t_{k+1}) + \sum_{j\in N^i}\nabla_{\mathbf{x}^i}g^{i,j}(\mathbf{x}^i_{k+1|k}, \mathbf{x}^j_{k+1|k}; t_{k+1}) \tag{3.23}$$

is also computed at node i, Lemma 1 asserts that the component $\mathbf{c}^i_{k+1,(K)}$ can indeed be calculated using local operations, which means that the iterative application of (3.15) can be executed in a distributed fashion. Consequently, node i computes $\mathbf{x}^i_{k+1}$ by implementing the following local descent:

$$\mathbf{x}^i_{k+1} = \mathbf{x}^i_{k+1|k} - \gamma\mathbf{c}^i_{k+1,(K)}. \tag{3.24}$$

We call DeNSP as the decentralized Newton method with a backward Euler prediction that uses descent method [cf.,(3.24)] and the prediction step [cf.,(3.14)]. The algorithm is summarized as Algorithm 1.

Algorithm 1: Decentralized newton method with a simple prediction (DeNSP) at node i.

```
Input: The local variable $\mathbf{x}_k^i$, the approximation level K, the step size $\gamma$.
for $t_k (k = 0, 1, 2, ...)$ do
    Predict the next trajectory using the prior information
    if $k > 0$
        $\mathbf{x}_{k+1|k}^i = 2\mathbf{x}_k^i - \mathbf{x}_{k-1}^i$
    else
        $\mathbf{x}_{k+1|k}^i = \mathbf{x}_0^i$
    end if
    Initialize the sequence of corrected variables $\hat{\mathbf{x}}_{k+1}^{i,0} = \mathbf{x}_{k+1|k}^i$
    Exchange the variable $\hat{\mathbf{x}}_{k+1}^{i,\eta}$ with neighbors $j \in N^i$
    Observe $F^i(.; t_{k+1})$, compute $\nabla_{\mathbf{x}} F^i(\mathbf{x}_{k+1|k}; t_k + 1)$ [cf.,(3.23)]
    Compute matrices $\mathbf{D}_{k+1|k}^{ii}$ and $\mathbf{B}_{k+1|k}^{ij}$, $j \in N^i$ as
        $\mathbf{D}_{k+1|k}^{ii} := \nabla_{\mathbf{x}^i\mathbf{x}^i} f^i(\mathbf{x}_{k+1|k}^i; t_{k+1})$
            $+\nabla_{\mathbf{x}^i\mathbf{x}^i} g^{i,i}(\mathbf{x}_{k+1|k}^i; t_{k+1})$
            $+ \sum_{j \in N^i} \nabla_{\mathbf{x}^i\mathbf{x}^i} g^{i,j}(\mathbf{x}_{k+1|k}^i, \mathbf{x}_{k+1|k}^j; t_{k+1})$
        $\mathbf{B}_{k+1|k}^{ij} = -\nabla_{\mathbf{x}^i\mathbf{x}^j} g^{i,j}(\mathbf{x}_{k+1|k}^i, \mathbf{x}_{k+1|k}^j; t_{k+1})$
    Compute:
        $\mathbf{c}_{k+1,(0)}^i = -(\mathbf{D}_{k+1|k}^{ii})^{-1} \nabla_{\mathbf{x}} F^i(\mathbf{x}_{k+1|k}; t_k + 1)$
    for $\tau = 0, 1, 2, ..., K - 1$ do
        Exchange correction step $\mathbf{c}_{k,(\tau)}^i$ with neighboring nodes $j \in N^i$
            $\mathbf{c}_{k+1,(\tau+1)}^i = -(\mathbf{D}_{k+1|k}^{ii})^{-1} \left( \sum_{j \in N^i} \mathbf{B}_{k+1|k}^{ij} \mathbf{c}_{k+1,(\tau)}^j + \nabla_{\mathbf{x}} F^i(\mathbf{x}_{k+1|k}; t_{k+1}) \right)$
    end for
    Correct the trajectory prediction
        $\mathbf{x}_{k+1}^i = \hat{\mathbf{x}}_{k+1|k}^i + \gamma \mathbf{c}_{k+1,(K')}^i$
end for
Output: The corrected variable $\mathbf{x}_{k+1}^i$
```

4. Convergence analysis

In this section, we examine the convergence of the algorithms introduced in Section III. Our convergence analysis demonstrates that the discrepancy between the optimal solution $\mathbf{x}^*(t_k)$ and the computed solution $\mathbf{x}_k$ is ultimately upper bounded. Specifically, the error is predominantly influenced by the sampling time h. Throughout this paper, we adopt the following assumption regarding the objective function:

Assumption 1. *Each node's own function f^i is twice differentiable, and the eigenvalues of the local*

Hessians matrix $\nabla_{\mathbf{x}^i\mathbf{x}^i} f^i(\mathbf{x}^i;t)$ *are bounded with positive constants* $[m_1, M_1]$*, i.e.,*

$$m_1 I \preceq \nabla_{\mathbf{x}^i\mathbf{x}^i} f^i(\mathbf{x}^i;t) \preceq M_1 I. \tag{4.1}$$

Assumption 2. *Functions* $g^{i,i}(\mathbf{x}^i;t)$ *and* $g^{i,j}(\mathbf{x}^i,\mathbf{x}^j;t)$ *are twice differentiable and the eigenvalues of the aggregate function Hessian are bounded* $[0, L_1]$*, the constant* $L_1 < \infty$*,*

$$0 \preceq \nabla_{\mathbf{x}\mathbf{x}} g(\mathbf{x};t) \preceq L_1 I. \tag{4.2}$$

Assumption 3. *The function* $F(\mathbf{x};t)$ *regarding all the derivatives of* $\mathbf{x} \in \mathbb{R}^{np}$ *and* $t \geq 0$ *are bounded,*

$$\begin{aligned} &\|\nabla_{t\mathbf{x}} F(\mathbf{x};t)\| \leq D_0, \quad \|\nabla_{\mathbf{x}\mathbf{x}\mathbf{x}} F(\mathbf{x};t)\| \leq D_1, \\ &\|\nabla_{\mathbf{x}t\mathbf{x}} F(\mathbf{x};t)\| \leq D_2, \quad \|\nabla_{tt\mathbf{x}} F(\mathbf{x};t)\| \leq D_3, \quad \|\nabla_{\mathbf{x}} F\| \leq D_4. \end{aligned} \tag{4.3}$$

From the abounds of Hessians $\nabla_{\mathbf{x}\mathbf{x}} f(\mathbf{x};t)$ and $\nabla_{\mathbf{x}\mathbf{x}} g(\mathbf{x};t)$ in Assumptions 1 and 2 respectively, the Hessian of global cost $\nabla_{\mathbf{x}\mathbf{x}} F(\mathbf{x};t)$ uniformly satisfies

$$m_1 I \preceq \nabla_{\mathbf{x}\mathbf{x}} F(\mathbf{x};t) \preceq (L_1 + M_1) I. \tag{4.4}$$

These conditions ensure that the problem formulated in (2.1) is strongly convex, and they also ensure the invertibility of the Hessian matrix. To refine the contribution of our paper, we impose bounds on the higher-order derivatives of F, as stated in Assumption 3. A similar assumption has been utilized in previous works [6, 10, 18].

We turn to analyzing the DeNSP algorithm. As per the definition of $\mathbf{D}_{k+1|k}$ and $\mathbf{B}_{k+1|k}$ in (3.18), the matrix $\mathbf{D}_{k+1|k}^{-1/2}\mathbf{B}_{k+1|k}\mathbf{D}_{k+1|k}^{-1/2}$ is positive semidefinite, and its eigenvalues are upper-bounded by a constant $\rho < 1$, as shown in Proposition 2 of [17].

$$0 \preceq \mathbf{D}_{k+1|k}^{-1/2}\mathbf{B}_{k+1|k}\mathbf{D}_{k+1|k}^{-1/2} \preceq \rho I, \tag{4.5}$$

where $\rho = (1 + \frac{2m_1}{L_1})^{-1} < 1$.

In the following theorem, we establish that the sequence generated by the DeNSP algorithm asymptotically converges to a neighborhood of the optimal trajectory whose radius depends on the sampling period h.

Proposition 1. *The norm of the difference between prediction* $\mathbf{x}_{k+1|k}$ *and the optimal solution* $\mathbf{x}^*_{k+1}$ *is upper bounded by*

$$\|\mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1}\| \leq \delta \|\mathbf{x}_k - \mathbf{x}^*_k\| + h^2 \Delta, \tag{4.6}$$

where $\delta := 1 + h\Big(\frac{D_1(D_0+\gamma D_4)}{m_1^2} + \frac{D_2+\gamma(L_1+M_1)}{m_1}\Big)$*, and* $\Delta = \frac{D_1(D_0+\gamma D_4)+(D_0+\gamma(L_1+M_1))}{m_1^3} + \frac{(D_0+\gamma D_4)(2D_2+\gamma m_1)}{m_1^2} + \frac{\gamma D_0+D_3}{m_1}$.

Proof. See Supplementary A. □

Theorem 1. *Under Assumptions 1–3, fixing* K *as the level of Hessian inverse approximation for the correction step, there exist bounds* $\bar{K}$*,* $\bar{h}$*, and* $\bar{R}$*, such that if the sampling period* h *is selected to satisfy*

$h \leq \bar{h}$, K is chosen to be $K \geq \bar{K}$, and the initial optimality gap fulfills $\|\mathbf{x}_0 - \mathbf{x}^*(t_0)\| \leq \bar{R}$, then $\mathbf{x}_{k+1}$ converges to the optimal trajectory $\mathbf{x}^*(t_{k+1})$ within a bounded error.

$$\lim_{k\to\infty} \sup \|\mathbf{x}_{k+1} - \mathbf{x}^*(t_{k+1})\| = O(h^2\rho^{K+1}(1-\gamma)) + O(h^4), \tag{4.7}$$

specially, when approximation level K is chosen sufficiently large and $\gamma = 1$, then the

$$\lim_{k\to\infty} \sup \|\mathbf{x}_{k+1} - \mathbf{x}^*(t_{k+1})\| = O(h^4). \tag{4.8}$$

Proof. See Appendix B. □

Theorem 1 indicates that the error bound described in (4.7) for the DeNSP algorithm is primarily influenced by the sampling period h. In the worst-case scenario, the asymptotic error floor is expected to be on the order of $O(h^2)$. However, according to the result $\lim_{k\to\infty} \sup \|\mathbf{x}_{k+1} - \mathbf{x}^*(t_{k+1})\| = O(h^4)$, it is evident that under certain conditions, the error bound can be further improved. Specifically, if the approximation level K is chosen to be sufficiently large, the DeNSP algorithm has the potential to achieve a much tighter tracking performance, reducing the asymptotic error to the order of $O(h^4)$. This suggests that by carefully selecting the parameter K, the algorithm can provide significantly better accuracy in tracking the optimal solution over time. Meanwhile, reducing the sampling interval provides greater assistance in improving convergence accuracy. Furthermore, since the DeNSP algorithm omits the calculation of the Hessian matrix inverse in the prediction step, the computational complexity of the prediction step is $O(p)$, while other algorithms such as DPC-N and DAPC-N require the calculation of the Hessian inverse in the prediction step, ignoring the complexity of gradient calculation, resulting in a computational complexity of $O(p^3)$. In the correction step, the computational complexity of several algorithms is consistent.

5. Simulation

In this section, we provide a numerical example to demonstrate the effectiveness of the DeNSP algorithm. We consider a resource allocation problem in a network of interconnected devices, as discussed in Section II.

We solve the problem described by (2.2) using an approximate augmented Lagrangian method, as outlined in (2.3). The local objective functions associated with sensor i are defined as follows:

$$f^i(\mathbf{x}^i; t) := \frac{1}{2}\Big(\mathbf{x}^i - \cos(wt)\Big)^2 + k\sum_{l=1}^{p} \log[1 + \exp b^{i,l}(x^{i,l} - d(t))]. \tag{5.1}$$

In our simulation, we consider the case where decisions are the variables $\mathbf{x}^i \in \mathbb{R}^p$, $p = 10$, and set the scalar parameters as $w = 0.2\pi$, $k = 0.1$, $b^{i,l} \sim \mathcal{U}^1_{[-2,2]}$ and $d(t) = \cos(\omega t)$. We consider there are $n = 50$ nodes in a wireless network which the nodes are randomly distributed in the square $[-1, 1]^2$ and they can only communicate with each other if they are closer than a range of $r = 2.5\sqrt{2}/\sqrt{n}$. Then the nodes generate a network with l links. We set the $\boldsymbol{b} = 0$ in (2.3) and the tuning parameter $\beta = \sqrt{20}$. We compare the DeNSP algorithm with DPC-N algorithm and DAPC-N algorithm mentioned in [6]. The constant step size of DPC-N and DAPC-N algorithm is set to $\gamma = 1$.

From convergence result (4.7), it can be seen that the convergence accuracy is affected by the number of iterations and sampling intervals in the correction step. In order to compare the advantages with other algorithms, we set a fixed and commonly used sampling interval $h = 0.1$. Further observe the impact of the number of iterations in the calibration process on convergence accuracy. In Figure 1, we run different algorithms and show the convergence performance between different algorithms. When the approximation level or the number of communication rounds K is set to 3, the tracking performance of these algorithms is comparable, showing little to no significant difference in their effectiveness. However, as K is increased to 5, the DeNSP algorithm demonstrates a clear advantage, achieving superior tracking accuracy and responsiveness. This suggests that the DeNSP algorithm benefits more from additional communication rounds, which enhances its ability to track the time-varying optimal solution more precisely, making it particularly effective in scenarios where higher precision is required. Further more, the DPC-N algorithm and the DAPC-N algorithm need to calculate of the inverse of the Hessian matrix, with the complexity typically represented as $O(p^3)$, where p is the matrix's dimension, and calculate mixed partial derivatives in the prediction step by communicating with neighbor nodes. In contrast, our algorithm can simply complete the prediction step using the previous information which can reduce computation time greatly.

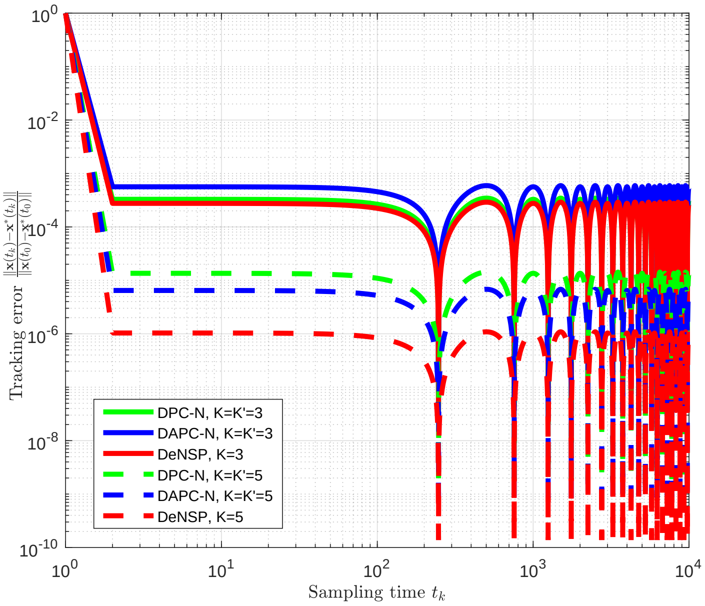

Figure 1. Error with respect to the sampling time t_k for different algorithms.

Although the prediction correction algorithm proposed in this article simplifies the prediction steps, the correction steps still require a large amount of computation on Hessian matrix. Furthermore, for some optimization problems involving integer decision variables, the algorithm proposed in this paper is difficult to solve and requires further research.

6. Conclusions

In this paper, we propose the backward Euler prediction step for the problem of the unconstrained distributed optimization. Though the theoretical analysis , the convergence accuracy of the proposed algorithm can reach $O(h^2) \sim O(h^4)$. And compared with the DPC-N algorithm and DAPC-N algorithm, our algorithm need not compute the inverse of the Hessian of the cost function, which can save computing time. Finally, we verify the theoretical results via a numerical example. In future research, we will further simplify the calculation of the Hessian matrix in the calibration step.

Author contributions

Zhuo Sun: Conceptualization, Methodology, Writing-review & editing; Huaiming Zhu: Formal analysis, Investigation, Software, Writing-original draft; Haotian Xu: Investigation, Writing-review & editing. All authors have read and approved the final version of the manuscript for publication.

Conflict of interest

The authors declare no conflict of interest.

Supported Foundation

This work is supported in part by the National Natural Science Foundation of China $(61304179, 71431001, 71831002, 72211540399)$; the Humanity and Social Science Youth Foundation of the Ministry of Education $(19YJC630151)$; the International Association of Maritime Universities (20200205_AMC); the Natural Science Foundation of Liaoning Province $(2020 - HYLH - 32)$; the Dalian Science and Technology Innovation Fund $(2020JJ26GX023)$

References

1. J. S. Pan, A. Q. Tian, V. Snášel, L. Kong, S. C. Chu, Maximum power point tracking and parameter estimation for multiple-photovoltaic arrays based on enhanced pigeon-inspired optimization with taguchi method, *Energy*, **251** (2022), 123863. https://doi.org/10.1016/j.energy.2022.123863

2. A. Q. Tian, X. Y. Wang, H. Xu, J. S. Pan, V. Snášel, H. X. Lv, Multi-objective optimization model for railway heavy-haul traffic: Addressing carbon emissions reduction and transport efficiency improvement, *Energy*, **294** (2024), 130927. https://doi.org/10.1016/j.energy.2024.130927

3. A. Simonetto, E. Dall'Anese, S. Paternain, G. Leus, G. B. Giannakis, Time-varying convex optimization: Time-structured algorithms and applications, *Proc. IEEE*, **108** (2020), 2032–2048. http://dx.doi.org/10.1109/JPROC.2020.3003156

4. Q. Li, Y. Liao, K. Wu, L. Zhang, J. Lin, M. Chen, J. M. Guerrero, et al., Parallel and distributed optimization method with constraint decomposition for energy management of microgrids, *Proc. IEEE*, **12** (2021), 4627–4640. http://dx.doi.org/10.1109/TSG.2021.3097047

5. S. Hosseini, A. Chapman, M. Mesbahi, Online distributed convex optimization on dynamic networks, *IEEE Trans. Autom. Control*, **61** (2016), 3545–3550. http://dx.doi.org/10.1109/TAC.2016.2525928

6. A. Simonetto, A. Koppel, A. Mokhtari, G. Leus, A. Ribeiro, Decentralized prediction-correction methods for networked time-varying convex optimization, *IEEE Trans. Autom. Control*, **62** (2017), 5724–5738. http://dx.doi.org/10.1109/TAC.2017.2694611

7. A. Simonetto, Dual prediction-correction methods for linearly constrained time-varying convex programs, *IEEE Trans. Autom. Control*, **64** (2018), 3355–3361. http://dx.doi.org/10.1109/TAC.2018.2877682

8. A. Q. Tian, F. F. Liu, H. X. Lv, Snow geese algorithm: A novel migration-inspired meta-heuristic algorithm for constrained engineering optimization problems, *Appl. Math. Model.*, **126** (2024), 327–347. https://doi.org/10.1016/j.apm.2023.10.045

9. X. Li, L. Xie, N. Li, A survey on distributed online optimization and game, *Arxiv Prep.*, 2022. https://doi.org/10.48550/arXiv.2205.00473

10. A. Simonetto, E. Dall'Anese, Prediction-correction algorithms for time-varying constrained optimization, *IEEE Trans. Signal Process.*, **65** (2017), 5481–5494. http://dx.doi.org/10.1109/TSP.2017.2728498

11. S. Qu, Y. Zhou, Y. Ji, Z. Dai, Z. Wang, Robust maximum expert consensus modeling with dynamic feedback mechanism under uncertain environments, *J. Ind. Manag. Optim.*, **12** (2024), 4627–4640. http://dx.doi.org/10.3934/jimo.2024093

12. S. Bittanti, F. A. Cuzzola, A mixed gh2/h approach for stabilization and accurate trajectory tracking of unicycle-like vehicles, *Int. J. Control*, **74** (2001), 880–888. https://doi.org/10.1080/00207170110037164

13. Y. Tang, Time-varying optimization and its application to power system operation, *California Instit. Tech.*, (2019). http://dx.doi.org/10.7907/6N9W-3J20

14. A. Y. Popkov, Gradient methods for nonstationary unconstrained optimization problems, *Autom. Remote Control*, **66** (2005), 883–891. https://doi.org/10.1007/s10513-005-0132-z

15. A. L. Dontchev, M. I. Krastanov, R. T. Rockafellar, V. M. Veliov, An euler-newton continuation method for tracking solution trajectories of parametric variational inequalities, *SIAM J. Control Optim.*, **51** (2013), 1823–1840. https://doi.org/10.1137/120876915

16. M. Fazlyab, S. Paternain, V. M. Preciado, A. Ribeiro, Prediction-correction interior-point method for time-varying convex optimization, *IEEE Trans. Autom. Control*, **63** (2017), 1973–1986. http://dx.doi.org/10.1109/TAC.2017.2760256

17. A. Mokhtari, Q. Ling, A. Ribeiro, Network newton-part i: Algorithm and convergence, 2015, *Arxiv Prep.*, (2015). https://doi.org/10.48550/arXiv.1504.06017

18. A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, A. Ribeiro, A class of prediction-correction methods for time-varying convex optimization, *IEEE Trans. Signal Process.*, **64** (2016), 4576–4591. http://dx.doi.org/10.1109/TSP.2016.2568161

19. P. Pedregal, *Introduction to optimization*, Springer, 2004. http://dx.doi.org/10.1007/b97412

20. V. M. Zavala, M. Anitescu, Real-time nonlinear optimization as a generalized equation, *SIAM J. Control Optim.*, **48** (2010), 5444–5467. https://doi.org/10.1137/090762634

21. A. Mokhtari, Q. Ling, A. Ribeiro, Network newton distributed optimization methods, *IEEE Trans. Signal Process.*, **65** (2017), 146–161. https://doi.org/10.1109/TSP.2016.2617829

22. Q. Alfio, S. Riccardo, S. Fausto, Numerical mathematics, *Springer Sci. Busin. Media*, **37** (2010). https://doi.org/10.1007/978-1-4612-4442-4

Supplementary

6.1. Proof of Prosition 1

In order to facilitate the presentation in the proof process, we define the following abbreviations

$$\begin{aligned}\nabla_{\mathbf{x}}F &= \nabla_{\mathbf{x}}F(\mathbf{x}_{k+1|k};t_{k+1}), \quad \nabla_{\mathbf{x}}F^* = \nabla_{\mathbf{x}}F(\mathbf{x}^*_{k+1};t_{k+1}),\\ \nabla_{\mathbf{xx}}F &= \nabla_{\mathbf{xx}}F(\mathbf{x}_{k+1|k};t_{k+1}), \quad \nabla_{\mathbf{xx}}F^* = \nabla_{\mathbf{xx}}F(\mathbf{x}^*_{k+1};t_{k+1}),\\ \nabla_{t\mathbf{x}}F &= \nabla_{t\mathbf{x}}F(\mathbf{x}_{k+1|k};t_{k+1}), \quad \nabla_{t\mathbf{x}}F^* = \nabla_{t\mathbf{x}}F(\mathbf{x}^*_{k+1};t_{k+1}).\end{aligned} \tag{6.1}$$

We define the dynamical system $\dot{\mathbf{x}}(t) = \mathcal{F}(\mathbf{x};t)$, observing the continuous dynamical system in (3.6), we know that $\mathcal{F}(\mathbf{x};t)$ is given by

$$\mathcal{F}(\mathbf{x};t) = -\nabla_{\mathbf{xx}}F(\mathbf{x};t)^{-1}\nabla_{t\mathbf{x}}F(\mathbf{x};t) - \gamma\nabla_{\mathbf{xx}}F(\mathbf{x};t)^{-1}\nabla_{\mathbf{x}}F(\mathbf{x};t). \tag{6.2}$$

For $\mathbf{x}_{k-1}$, use Taylor expansion at $\mathbf{x}_k$

$$\mathbf{x}_{k-1} = \mathbf{x}_k - h\mathcal{F}(\mathbf{x};t_k) + \frac{h^2}{2}\frac{d}{dt}\mathcal{F}(\mathbf{x};s_1), \quad s_1 \in [t_{k-1},t_k]. \tag{6.3}$$

According to (6.3), we have

$$\mathcal{F}(\mathbf{x};t_k) = \frac{\mathbf{x}_k - \mathbf{x}_{k-1}}{h} + \frac{h}{2}\frac{d}{dt}\mathcal{F}(\mathbf{x};s_1). \tag{6.4}$$

According to (6.4), the prediction step (3.12) can be written as

$$\mathbf{x}_{k+1|k} = \mathbf{x}_k + h\mathcal{F}(\mathbf{x};t_k) - \frac{h^2}{2}\frac{d\mathcal{F}(\mathbf{x};s_1)}{dt}. \tag{6.5}$$

Taylor expansion of $\mathbf{x}^*_{k+1}$ have

$$\mathbf{x}^*_{k+1} = \mathbf{x}^*_k + h\mathcal{F}(\mathbf{x}^*;t_k) + \frac{h^2}{2}\frac{d\mathcal{F}(\mathbf{x}^*;s_2)}{dt} \quad s_2 \in [t_k,t_{k+1}]. \tag{6.6}$$

Subtracting (6.6) from (6.5) yields

$$||\mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1}|| = ||\mathbf{x}_k - \mathbf{x}^*_k|| + h(\mathcal{F}(\mathbf{x};t_k) - \mathcal{F}(\mathbf{x}^*;t_k)) - \frac{h^2}{2}\Big(\frac{d\mathcal{F}(\mathbf{x}^*;s_1)}{dt} + \frac{d\mathcal{F}(\mathbf{x}^*;s_2)}{dt}\Big). \tag{6.7}$$

Applying the triangle inequality to (6.7) gives

$$||\mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1}|| \le ||\mathbf{x}_k - \mathbf{x}^*_k|| + h\,||\mathcal{F}(\mathbf{x};t_k) - \mathcal{F}(\mathbf{x}^*;t_k)|| + \frac{h^2}{2}\left\|\frac{d\mathcal{F}(\mathbf{x}^*;s_1)}{dt} + \frac{d\mathcal{F}(\mathbf{x}^*;s_2)}{dt}\right\|. \tag{6.8}$$

For the second term of (6.8), combined with the formula (6.2), we have

$$\begin{aligned}||\mathcal{F}(\mathbf{x};t_k) - \mathcal{F}(\mathbf{x}^*;t_k)|| &= ||-\nabla_{\mathbf{xx}}F^{-1}\nabla_{t\mathbf{x}}F - \gamma\nabla_{\mathbf{xx}}F^{-1}\nabla_{\mathbf{x}}F - \Big(-[\nabla_{\mathbf{xx}}F^*]^{-1}\nabla_{t\mathbf{x}}F^* - \gamma[\nabla_{\mathbf{xx}}F^*]^{-1}\nabla_{\mathbf{x}}F^*\Big)||\\ &\le ||[\nabla_{\mathbf{xx}}F^*]^{-1}\nabla_{t\mathbf{x}}F^* - \nabla_{\mathbf{xx}}F^{-1}\nabla_{t\mathbf{x}}F|| + \gamma||[\nabla_{\mathbf{xx}}F^*]^{-1}\nabla_{\mathbf{x}}F^* - \nabla_{\mathbf{xx}}F^{-1}\nabla_{\mathbf{x}}F||.\end{aligned} \tag{6.9}$$

For the first term of (6.9), add and subtract $[\nabla_{\mathbf{xx}}F^*]^{-1}\nabla_{t\mathbf{x}}F$ at the same time, and use the triangle inequality that

$$\begin{aligned}\|[\nabla_{\mathbf{xx}}F^*]^{-1}\nabla_{t\mathbf{x}}F^* - \nabla_{\mathbf{xx}}F^{-1}\nabla_{t\mathbf{x}}F\| \leq{}& \|\nabla_{\mathbf{xx}}F^{-1}\nabla_{t\mathbf{x}}F - [\nabla_{\mathbf{xx}}F^*]^{-1}\nabla_{t\mathbf{x}}F\| \\ &+\|[\nabla_{\mathbf{xx}}F^*]^{-1}\nabla_{t\mathbf{x}}F - [\nabla_{\mathbf{xx}}F^*]^{-1}\nabla_{t\mathbf{x}}F^*\|.\end{aligned} \tag{6.10}$$

The RHS of (6.10) can be bounded as

$$\|[\nabla_{\mathbf{xx}}F^*]^{-1}\nabla_{t\mathbf{x}}F^* - \nabla_{\mathbf{xx}}F^{-1}\nabla_{t\mathbf{x}}F\| \leq D_0\|\nabla_{\mathbf{xx}}F^{-1} - [\nabla_{\mathbf{xx}}F^*]^{-1}\| + \frac{1}{m_1}\|\nabla_{t\mathbf{x}}F - \nabla_{t\mathbf{x}}F^*\|, \tag{6.11}$$

where Assumption 3 is used.

Additionally, the bound on $\|\nabla_{\mathbf{xx}}F^{-1} - [\nabla_{\mathbf{xx}}F^*]^{-1}\|$ can be given as

$$\begin{aligned}\|\nabla_{\mathbf{xx}}F^{-1} - [\nabla_{\mathbf{xx}}F^*]^{-1}\| &= \|[\nabla_{\mathbf{xx}}F^*]^{-1}(\nabla_{\mathbf{xx}}F - \nabla_{\mathbf{xx}}F^*)\nabla_{\mathbf{xx}}F^{-1}\| \\ &\leq \frac{1}{m_1^2}\|\nabla_{\mathbf{xx}}F - \nabla_{\mathbf{xx}}F^*\|.\end{aligned} \tag{6.12}$$

For the second term of (6.11) and (6.12), applying the mean value theorem gives

$$\begin{aligned}\|\nabla_{t\mathbf{x}}F - \nabla_{t\mathbf{x}}F^*\| &= \|\nabla_{\mathbf{x}t\mathbf{x}}F(\tilde{\mathbf{x}}_1; t_k)(\mathbf{x}_k - \mathbf{x}_k^*)\| \\ &\leq \|\nabla_{\mathbf{x}t\mathbf{x}}F(\tilde{\mathbf{x}}_1; t_k)\|\|\mathbf{x}_k - \mathbf{x}_k^*\| \\ &\leq D_2\|\mathbf{x}_k - \mathbf{x}_k^*\|,\end{aligned} \tag{6.13}$$

$$\begin{aligned}\|\nabla_{\mathbf{xx}}F^{-1} - [\nabla_{\mathbf{xx}}F^*]^{-1}\| &\leq \frac{1}{m_1^2}\|\nabla_{\mathbf{xxx}}F(\tilde{\mathbf{x}}_2; t_k)\left(\mathbf{x}_k - \mathbf{x}_k^*\right)\| \\ &\leq \frac{1}{m_1^2}\|\nabla_{\mathbf{xxx}}F(\tilde{\mathbf{x}}_2; t_k)\|\|\mathbf{x}_k - \mathbf{x}_k^*\| \\ &\leq \frac{D_1}{m_1^2}\|\mathbf{x}_k - \mathbf{x}_k^*\|,\end{aligned} \tag{6.14}$$

where $\tilde{\mathbf{x}}_2$ and $\tilde{\mathbf{x}}_2$ are variables between $\mathbf{x}_k$ and $\mathbf{x}_k^*$.

Substituting (6.13) and (6.14) into (6.11), get

$$\|[\nabla_{\mathbf{xx}}F^*]^{-1}\nabla_{t\mathbf{x}}F^* - \nabla_{\mathbf{xx}}F^{-1}\nabla_{t\mathbf{x}}F\| \leq \left(\frac{D_0D_1}{m_1^2} + \frac{D_2}{m_1}\right)\|\mathbf{x}_k - \mathbf{x}_k^*\|. \tag{6.15}$$

For the second term of (6.9), add and subtract $[\nabla_{\mathbf{xx}}F^*]^{-1}\nabla_{\mathbf{x}}F$ at the same time, get

$$\begin{aligned}&\|[\nabla_{\mathbf{xx}}F^*]^{-1}\nabla_{\mathbf{x}}F^* - [\nabla_{\mathbf{xx}}F^*]^{-1}\nabla_{\mathbf{x}}F + [\nabla_{\mathbf{xx}}F^*]^{-1}\nabla_{\mathbf{x}}F - \nabla_{\mathbf{xx}}F^{-1}\nabla_{\mathbf{x}}F\| \\ &\leq \frac{1}{m_1}\|\nabla_{\mathbf{x}}F^* - \nabla_{\mathbf{x}}F\| + D_4\|[\nabla_{\mathbf{xx}}F^*]^{-1} - \nabla_{\mathbf{xx}}F^{-1}\|.\end{aligned} \tag{6.16}$$

For the first term of (6.16), apply the mean value theorem gives

$$\begin{aligned}\|\nabla_{\mathbf{x}}F^* - \nabla_{\mathbf{x}}F\| &= \|\nabla_{\mathbf{xx}}F(\tilde{\mathbf{x}}_3; t)(\mathbf{x}_k - \mathbf{x}_k^*)\| \\ &\leq \|\nabla_{\mathbf{xx}}F(\tilde{\mathbf{x}}_3; t)\|\|\mathbf{x}_k - \mathbf{x}_k^*\| \\ &\leq (L_1 + M_1)\|\mathbf{x}_k - \mathbf{x}_k^*\|,\end{aligned} \tag{6.17}$$

where $\tilde{\mathbf{x}}_3$ is a variable between $\mathbf{x}_k$ and $\mathbf{x}_k^*$.

Substituting (6.14) and (6.17) into (6.16) gives

$$\|[\nabla_{\mathbf{xx}}F^*]^{-1}\nabla_{\mathbf{x}}F^* - \nabla_{\mathbf{xx}}F^{-1}\nabla_{\mathbf{x}}F\| \leq \Big(\frac{L_1 + M_1}{m_1} + \frac{D_1 D_4}{m_1^2}\Big)\|\mathbf{x}_k - \mathbf{x}_k^*\|. \tag{6.18}$$

Substituting (6.15) and (6.18) into (6.9) gives

$$\|\mathcal{F}(\mathbf{x}; t_k) - \mathcal{F}(\mathbf{x}^*; t_k)\| \leq \Big(\frac{D_1(D_0 + \gamma D_4)}{m_1^2} + \frac{D_2 + \gamma(L_1 + M_1)}{m_1}\Big)\|\mathbf{x}_k - \mathbf{x}_k^*\|. \tag{6.19}$$

Substituting (6.19) into (6.8) leads to

$$\|\mathbf{x}_{k+1|k} - \mathbf{x}_k^*\| = \Big[1 + h\Big(\frac{D_1(D_0 + \gamma D_4)}{m_1^2} + \frac{D_2 + \gamma(L_1 + M_1)}{m_1}\Big)\Big]\|\mathbf{x}_k - \mathbf{x}_k^*\| + \frac{h^2}{2}\left\|\frac{d\mathcal{F}(\mathbf{x}^*; s_1)}{dt} + \frac{d\mathcal{F}(\mathbf{x}^*; s_2)}{dt}\right\|. \tag{6.20}$$

For the second term of (6.20), applying the triangle inequality, leads to

$$\frac{h^2}{2}\left\|\frac{d}{dt}\mathcal{F}(\mathbf{x}; s_1) + \frac{d}{dt}\mathcal{F}(\mathbf{x}; s_2)\right\| \leq \frac{h^2}{2}\left\|\frac{d}{dt}\mathcal{F}(\mathbf{x}^*; s_1)\right\| + \frac{h^2}{2}\left\|\frac{d}{dt}\mathcal{F}(\mathbf{x}^*; s_2)\right\|. \tag{6.21}$$

The derivative of $\mathcal{F}(\mathbf{x}; t)$ is given by

$$\begin{aligned}\frac{d\mathcal{F}(\mathbf{x}(t); t)}{dt} &= \nabla_t\mathcal{F}(\mathbf{x}; t) + [\nabla_{\mathbf{x}}\mathcal{F}(\mathbf{x}; t)]\dot{\mathbf{x}} \\ &= \nabla_t\mathcal{F}(\mathbf{x}; t) + [\nabla_{\mathbf{x}}\mathcal{F}(\mathbf{x}; t)]\mathcal{F}(\mathbf{x}; t).\end{aligned} \tag{6.22}$$

Applying the triangle inequality to (6.2) yields

$$\begin{aligned}\|\mathcal{F}(\mathbf{x}; t)\| &\leq \|[\nabla_{\mathbf{xx}}F(\mathbf{x}; t)]^{-1}\|\|\nabla_{t\mathbf{x}}F(\mathbf{x}; t)\| + \gamma\|[\nabla_{\mathbf{xx}}F(\mathbf{x}; t)]^{-1}\|\|\nabla_{\mathbf{x}}F(\mathbf{x}; t)\| \\ &\leq \frac{D_0 + \gamma D_4}{m_1},\end{aligned} \tag{6.23}$$

and to (6.22) yields

$$\left\|\frac{d\mathcal{F}(\mathbf{x}; t)}{dt}\right\| \leq \|\nabla_t\mathcal{F}(\mathbf{x}; t)\| + \|[\nabla_{\mathbf{x}}\mathcal{F}(\mathbf{x}; t)]\mathcal{F}(\mathbf{x}; t)\|. \tag{6.24}$$

It follows from (6.2) that

$$\nabla_t\mathcal{F}(\mathbf{x}; t) = \nabla_t[-\nabla_{\mathbf{xx}}F(\mathbf{x}; t)^{-1}\nabla_{t\mathbf{x}}F(\mathbf{x}; t) - \gamma\nabla_{\mathbf{xx}}F(\mathbf{x}; t)^{-1}\nabla_{\mathbf{x}}F(\mathbf{x}; t)], \tag{6.25}$$

where

$$\begin{aligned}&\nabla_t\left[-\nabla_{\mathbf{xx}}F(\mathbf{x}; t)^{-1}\nabla_{t\mathbf{x}}F(\mathbf{x}; t) - \gamma\nabla_{\mathbf{xx}}F(\mathbf{x}; t)^{-1}\nabla_{\mathbf{x}}F(\mathbf{x}; t)\right] \\ &= \nabla_{\mathbf{xx}}F(\mathbf{x}; t)^{-2}\nabla_{t\mathbf{xx}}F(\mathbf{x}; t)\nabla_{t\mathbf{x}}F(\mathbf{x}; t) - \nabla_{\mathbf{xx}}F(\mathbf{x}; t)^{-1}\nabla_{tt\mathbf{x}}F(\mathbf{x}; t) \\ &+ \gamma\nabla_{\mathbf{xx}}F(\mathbf{x}; t)^{-2}\nabla_{t\mathbf{xx}}F(\mathbf{x}; t)\nabla_{\mathbf{x}}F(\mathbf{x}; t) - \gamma\nabla_{\mathbf{xx}}F(\mathbf{x}; t)^{-1}\nabla_{t\mathbf{x}}F(\mathbf{x}; t).\end{aligned} \tag{6.26}$$

Using the triangle inequality for (6.26)

$$\begin{aligned}\|\nabla_t\mathcal{F}(\mathbf{x};t)\| &= \|\nabla_t[-\nabla_{\mathbf{xx}}F(\mathbf{x};t)^{-1}\nabla_{t\mathbf{x}}F(\mathbf{x};t) - \gamma\nabla_{\mathbf{xx}}F(\mathbf{x};t)^{-1}\nabla_{\mathbf{x}}F(\mathbf{x};t)]\| \\ &\leq \|\nabla_{\mathbf{xx}}F(\mathbf{x};t)^{-2}\nabla_{t\mathbf{xx}}F(\mathbf{x};t)\nabla_{t\mathbf{x}}F(\mathbf{x};t)\| \\ &+ \|\nabla_{\mathbf{xx}}F(\mathbf{x};t)^{-1}\nabla_{tt\mathbf{x}}F(\mathbf{x};t)\| \\ &+ \gamma\|\nabla_{\mathbf{xx}}F(\mathbf{x};t)^{-2}\nabla_{t\mathbf{xx}}F(\mathbf{x};t)\nabla_{\mathbf{x}}F(\mathbf{x};t)\| \\ &+ \gamma\|\nabla_{\mathbf{xx}}F(\mathbf{x};t)^{-1}\nabla_{t\mathbf{x}}F(\mathbf{x};t)\| \\ &\leq \frac{D_0(D_2+\gamma D_4)}{{m_1}^2} + \frac{\gamma D_0 + D_3}{m_1},\end{aligned} \tag{6.27}$$

and we also can get

$$\begin{aligned}\|[\nabla_{\mathbf{x}}\mathcal{F}(\mathbf{x};t)]\mathcal{F}(\mathbf{x};t)\| &= \Big\|\Big(- \nabla_{\mathbf{xx}}F(\mathbf{x};t)^{-1}\nabla_{\mathbf{x}t\mathbf{x}}F(\mathbf{x};t) - \gamma \\ &+ \nabla_{\mathbf{xx}}F(\mathbf{x};t)^{-2}\nabla_{\mathbf{xxx}}F(\mathbf{x};t)\nabla_{t\mathbf{x}}F(\mathbf{x};t) \\ &+ \gamma\nabla_{\mathbf{xx}}F(\mathbf{x};t)^{-2}\nabla_{\mathbf{xxx}}F(\mathbf{x};t)\nabla_{\mathbf{x}}F(\mathbf{x};t)\Big)\mathcal{F}(\mathbf{x};t)\Big\| \\ &\leq \Big(\frac{D_1(D_0+\gamma(L_1+M_1))}{m_1^2} + \frac{D_2+\gamma m_1}{m_1}\Big)\Big(\frac{D_0+\gamma D_4}{m_1}\Big).\end{aligned} \tag{6.28}$$

It follows from (6.27) and (6.28), the (6.24) can be written as

$$\begin{aligned}\Big\|\frac{d\mathcal{F}(\mathbf{x};t)}{dt}\Big\| &\leq \|\nabla_t\mathcal{F}(\mathbf{x};t)\| + \|[\nabla_{\mathbf{x}}\mathcal{F}(\mathbf{x};t)]\mathcal{F}(\mathbf{x};t)\| \\ &\leq \frac{D_1(D_0+\gamma D_4)+(D_0+\gamma(L_1+M_1))}{m_1^3} + \frac{(D_0+\gamma D_4)(2D_2+\gamma m_1)}{m_1^2} + \frac{\gamma D_0+D_3}{m_1}.\end{aligned} \tag{6.29}$$

According to the (6.20) and (6.29), the (6.30) follows

$$\begin{aligned}\|\mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1}\| \leq &\Big[1 + h\Big(\frac{D_1(D_0+\gamma D_4)}{m_1^2} + \frac{D_2+\gamma(L_1+M_1)}{m_1}\Big)\Big]\|\mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1}\| \\ &+ h^2\left(\frac{D_1(D_0+\gamma D_4)+(D_0+\gamma(L_1+M_1))}{m_1^3} + \frac{(D_0+\gamma D_4)(2D_2+\gamma m_1)}{m_1^2} + \frac{\gamma D_0+D_3}{m_1}\right),\end{aligned} \tag{6.30}$$

which can be written as

$$\|\mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1}\| \leq \delta\|\mathbf{x}_k - \mathbf{x}^*_k\| + h^2\Delta, \tag{6.31}$$

where $\delta := 1 + h\Big(\frac{D_1(D_0+\gamma D_4)}{m_1^2} + \frac{D_2+\gamma(L_1+M_1)}{m_1}\Big)$, and $\Delta = \frac{D_1(D_0+\gamma D_4)+(D_0+\gamma(L_1+M_1))}{m_1^3} + \frac{(D_0+\gamma D_4)(2D_2+\gamma m_1)}{m_1^2} + \frac{\gamma D_0+D_3}{m_1}$.

The proof is completed.

6.2. *Proof of Theorem 1*

According to the Proposition 1, the norm of the difference between prediction $\mathbf{x}_{k+1|k}$ and optimal solution $\mathbf{x}^*_{k+1}$ is upper bounded by

$$\|\mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1}\| \leq \delta\|\mathbf{x}_k - \mathbf{x}^*_k\| + h^2\Delta, \tag{6.32}$$

where $\delta := 1 + h\Big(\frac{D_1(D_0+\gamma D_4)}{m_1^2} + \frac{D_2+\gamma(L_1+M_1)}{m_1}\Big)$, and $\Delta = \frac{D_1(D_0+\gamma D_4)+(D_0+\gamma(L_1+M_1))}{m_1^3} + \frac{(D_0+\gamma D_4)(2D_2+\gamma m_1)}{m_1^2} + \frac{\gamma D_0+D_3}{m_1}$. For $||\mathbf{x}_{k+1} - \mathbf{x}^*_{k+1}||$, according to (3.15), get

$$||\mathbf{x}_{k+1} - \mathbf{x}^*_{k+1}|| = ||\mathbf{x}_{k+1|k} - \gamma\mathbf{c}_{k+1,(K)} - \mathbf{x}^*_{k+1}||. \tag{6.33}$$

For convenience, we define

$$\mathbf{H}^{-1}_{k+1|k,(K)} = \mathbf{D}^{-\frac{1}{2}}_{k+1|k}\sum_{\tau=0}^{K}\Big(\mathbf{D}^{-\frac{1}{2}}_{k+1|k}\mathbf{B}_{k+1|k}\mathbf{D}^{-\frac{1}{2}}_{k+1|k}\Big)^{\tau}\mathbf{D}^{-\frac{1}{2}}_{k+1|k}. \tag{6.34}$$

According to the definition of $\mathbf{c}_{k+1,(K)}$ in (3.16), and substitute (6.34) into (6.33), (6.33) can be rewritten as

$$||\mathbf{x}_{k+1} - \mathbf{x}^*_{k+1}|| = ||\mathbf{x}_{k+1|k} - \gamma\mathbf{H}^{-1}_{k+1|k,(K)}\nabla_{\mathbf{x}}F - \mathbf{x}^*_{k+1}||. \tag{6.35}$$

Add and subtract $\gamma\nabla_{\mathbf{xx}}F^{-1}\nabla_{\mathbf{x}}F$ at the same time and use the triangle inequality, obtained

$$\begin{aligned}||\mathbf{x}_{k+1} - \mathbf{x}^*_{k+1}|| \leq\ & ||\mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1} - \gamma\nabla_{\mathbf{xx}}F^{-1}\nabla_{\mathbf{x}}F|| \\ & + \gamma||(\nabla_{\mathbf{xx}}F^{-1} - \mathbf{H}^{-1}_{k+1|k,(K)})\nabla_{\mathbf{x}}F||.\end{aligned} \tag{6.36}$$

For the first term on the RHS of (6.36), left multiply by $\nabla_{\mathbf{xx}}F$ and its inverse, and left factor out the Hessian inverse $\nabla_{\mathbf{xx}}F^{-1}$. Making use of the Cauchy-Schwartz inequality, the first term of RHS of (6.36) is bounded above as

$$\begin{aligned}& ||\mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1} - \gamma\nabla_{\mathbf{xx}}F^{-1}\nabla_{\mathbf{x}}F|| \\ & \leq (1-\gamma)||\mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1}|| + \gamma||\nabla_{\mathbf{xx}}F^{-1}||||\nabla_{\mathbf{xx}}F(\mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1}) - \nabla_{\mathbf{x}}F||.\end{aligned} \tag{6.37}$$

From Assumption 3, we know the upper bound of $||\nabla_{\mathbf{xx}}F^{-1}||$ is $1/m_1$. Additionally, according to the optimal condition $\nabla_{\mathbf{x}}F^*_{k+1} = 0$, then the (6.37) can be written as

$$\begin{aligned}||\mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1} - \gamma\nabla_{\mathbf{xx}}F^{-1}\nabla_{\mathbf{x}}F|| \leq\ & (1-\gamma)||\mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1}|| \\ & + \frac{\gamma}{m_1}||\nabla_{\mathbf{xx}}F(\mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1}) - (\nabla_{\mathbf{x}}F - \nabla_{\mathbf{x}}F^*_{k+1})||.\end{aligned} \tag{6.38}$$

Defining $\mathbf{r}_{k+1} = \mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1}$ and make $\mathbf{x} := \xi(\tau) = \mathbf{x}^*_{k+1} + \tau(\mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1})$, then

$$\begin{aligned}& \frac{\gamma}{m_1}||\nabla_{\mathbf{xx}}F(\mathbf{x}_{k+1} - \mathbf{x}^*_{k+1}) - (\nabla_{\mathbf{x}}F - \nabla_{\mathbf{x}}F^*_{k+1})|| \\ =& \frac{\gamma}{m_1}||\nabla_{\mathbf{xx}}F\mathbf{r}_{k+1} - (\nabla_{\mathbf{x}}F - \nabla_{\mathbf{x}}F^*_{k+1})|| \\ =& \frac{\gamma}{m_1}||\nabla_{\xi(\tau)\xi(\tau)}F\mathbf{r}_{k+1} - (\nabla_{\xi(\tau)}F - \nabla_{\xi(\tau)}F^*_{k+1})||.\end{aligned} \tag{6.39}$$

Because

$$\begin{aligned}\nabla_{\xi(\tau)}F - \nabla_{\xi(\tau)}F^*_{k+1} &= \int_{\mathbf{x}^*_{k+1}}^{\mathbf{x}_{k+1|k}} \nabla_{\xi(\tau)\xi(\tau)}F(\xi(\tau);t_{k+1})d\xi(\tau) \\ &= \int_0^1 \nabla_{\xi(\tau)\xi(\tau)}F(\xi(\tau);t_{k+1})\mathbf{r}_{k+1}d\tau.\end{aligned} \tag{6.40}$$

Substituting (6.40) into (6.39), derive

$$\begin{aligned}&\frac{\gamma}{m_1}\|\nabla_{\xi(\tau)\xi(\tau)}F(\mathbf{x}_{k+1|k}-x^*_{k+1})-(\nabla_{\xi(\tau)}F-\nabla_{\xi(\tau)}F^*_{k+1})\|\\&=\frac{\gamma}{m_1}\|\nabla_{\xi(\tau)\xi(\tau)}F\mathbf{r}_{k+1}-\int_0^1\nabla_{\xi(\tau)\xi(\tau)}F(\xi(\tau);t_{k+1})\mathbf{r}_{k+1}d\tau\|\\&=\frac{\gamma}{m_1}\|\mathbf{r}_{k+1}\int_0^1\nabla_{\xi(\tau)\xi(\tau)}F-\nabla_{\xi(\tau)\xi(\tau)}F(\xi(\tau);t_{k+1})d\tau\|\\&\leq\frac{\gamma}{m_1}\|\mathbf{r}_{k+1}\|\int_0^1\|\nabla_{\xi(\tau)\xi(\tau)}F-\nabla_{\xi(\tau)\xi(\tau)}F(\xi(\tau);t_{k+1})\|d\tau.\end{aligned}\tag{6.41}$$

For $\|\nabla_{\xi\xi}F-\nabla_{\xi\xi}F(\xi(\tau);t_{k+1})\|$, applying the mean value theorem and according to the assumption 1, derive

$$\begin{aligned}&\|\nabla_{\xi(\tau)\xi(\tau)}F-\nabla_{\xi(\tau)\xi(\tau)}F(\xi(\tau);t_{k+1})\|\\&\leq D_1\|\mathbf{x}_{k+1|k}-\xi(\tau)\|\\&\leq D_1\|(1-\tau)(\mathbf{x}_{k+1|k}-\mathbf{x}^*_{k+1})\|\\&\leq D_1(1-\tau)\|\mathbf{r}_{k+1}\|.\end{aligned}\tag{6.42}$$

Substituting (6.42) into (6.41), we have

$$\begin{aligned}&\frac{\gamma}{m_1}\|\nabla_{\mathbf{xx}}F(\mathbf{x}_{k+1|k}-x^*_{k+1})-(\nabla_{\mathbf{x}}F-\nabla_{\mathbf{x}}F^*_{k+1})\|\\&\leq\frac{\gamma}{m_1}\|\mathbf{r}_{k+1}\|\int_0^1 D_1(1-\tau)\|\mathbf{r}_{k+1}\|d\tau\\&\leq\frac{D_1}{2m_1}\|\mathbf{r}_{k+1}\|^2,\end{aligned}\tag{6.43}$$

substituting (6.43) into (6.37), we have

$$\|\mathbf{x}_{k+1|k}-\mathbf{x}^*_{k+1}-\gamma\nabla_{\mathbf{xx}}F^{-1}\nabla_{\mathbf{x}}F\|\leq\frac{D_1}{2m_1}\|\mathbf{x}_{k+1|k}-\mathbf{x}^*_{k+1}\|^2.\tag{6.44}$$

For $\|(\nabla_{\mathbf{xx}}F^{-1}-\mathbf{H}^{-1}_{k+1|k,(K)})\nabla_{\mathbf{x}}F\|$ which the second term in (6.36), we extract the Hessian inverse from it and use the triangle inequality, resulting in

$$\begin{aligned}\|(\nabla_{\mathbf{xx}}F^{-1}-\mathbf{H}^{-1}_{k+1|k,(K)})\nabla_{\mathbf{x}}F\|&\leq\|\nabla_{\mathbf{xx}}F^{-1}(\mathbf{I}-\nabla_{\mathbf{xx}}F\mathbf{H}^{-1}_{k+1|k,(K)})\nabla_{\mathbf{x}}F\|\\&\leq\frac{1}{m_1}\|\mathbf{I}-\nabla_{\mathbf{xx}}F\mathbf{H}^{-1}_{k+1|k,(K)}\|\|\nabla_{\mathbf{x}}F\|.\end{aligned}\tag{6.45}$$

According the Assumption 3, the norm $\|\nabla_{\mathbf{x}}F\|\leq D_4$, but we further reduce the upper bound of the norm, according to the optimality condition $\nabla_{\mathbf{x}}F^*=0$, get

$$\begin{aligned}\|\nabla_{\mathbf{x}}F\|&=\|\nabla_{\mathbf{x}}F-\nabla_{\mathbf{x}}F^*\|\\&\leq(L_1+M_1)\|\mathbf{x}_{k+1|k}-\mathbf{x}^*_{k+1}\|,\end{aligned}\tag{6.46}$$

where we have used the Lipschitz property of the gradients.

According the Lemma 3 in [17], the matrix $\mathbf{I} - \nabla_{\mathbf{xx}} F \mathbf{H}^{-1}_{k+1|k,(K)}$ can be simplified as

$$\mathbf{I} - \nabla_{\mathbf{xx}} F \mathbf{H}^{-1}_{k+1|k,(K)} = (\mathbf{B}_{k+1|k}\mathbf{D}^{-1}_{k+1|k})^{K+1}. \tag{6.47}$$

Since matrices $\mathbf{D}^{-1/2}_{k+1|k}\mathbf{B}_{k+1|k}\mathbf{D}^{-1/2}_{k+1|k}$ and $\mathbf{B}_{k+1|k}\mathbf{D}^{-1}_{k+1|k}$ are similar, the sets of eigenvalues of these two matrices are identical. Therefore, according to (4.5), we can get

$$0 \le \mathbf{B}_{k+1|k}\mathbf{D}^{-1}_{k+1|k} \le \rho I. \tag{6.48}$$

The bound of (6.47) is

$$||\mathbf{I} - \nabla_{\mathbf{xx}} F \mathbf{H}^{-1}_{k+1|k,(K)}|| \le \rho^{K+1}. \tag{6.49}$$

Substitute the (6.46) and (6.49) into (6.45), we can derive the (6.45) is bounded as

$$||(\nabla_{\mathbf{xx}} F^{-1} - \mathbf{H}^{-1}_{k+1|k,(K)})\nabla_{\mathbf{x}} F|| \le \frac{(L_1 + M_1)\rho^{K+1}}{m_1}||\mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1}||. \tag{6.50}$$

Applying the bounds (6.44) and (6.50) to the RHS of (6.36), get

$$||\mathbf{x}_{k+1} - \mathbf{x}^*_{k+1}|| \le \frac{\gamma D_1}{2m_1}||\mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1}||^2 + \psi||\mathbf{x}_{k+1|k} - \mathbf{x}^*_{k+1}||n \tag{6.51}$$

where $\psi = \frac{\gamma(L_1+M_1)\rho^{K+1}}{m_1} + 1 - \gamma$.

Substituting the (6.32) into the (6.51) yields

$$||\mathbf{x}_{k+1} - \mathbf{x}^*_{k+1}|| \le \alpha_2||\mathbf{x}_k - \mathbf{x}^*_k||^2 + \alpha_1||\mathbf{x}_k - \mathbf{x}^*_k|| + \alpha_0 n \tag{6.52}$$

where $\alpha_2 = \frac{\gamma D_1 \delta^2}{2m_1}$, $\alpha_1 = h^2\frac{D_1\delta\Delta}{m_1} + \psi\delta$, $\alpha_0 = h^4\frac{D_1\Delta^2}{2m_1} + h^2\psi\Delta$.

Let $0 < \tau_1 < 1$ be a positive scalar such that

$$\alpha_2||\mathbf{x}_k - \mathbf{x}^*_k||^2 + \alpha_1||\mathbf{x}_k - \mathbf{x}^*_k|| + \alpha_0 \le \tau_1||\mathbf{x}_k - \mathbf{x}^*_k|| + \alpha_0. \tag{6.53}$$

To guarantee the convergence of the error $||\mathbf{x}_k - \mathbf{x}^*_k||$ and (6.53) holds true for all k we require

$$\alpha_1 < \tau_1, \tag{6.54a}$$

$$||\mathbf{x}_0 - \mathbf{x}^*_0|| \le (\tau_1 - \alpha_1)/\alpha_2 = \bar{R}, \tag{6.54b}$$

$$\tau_1(\tau_1 - \alpha_1)/\alpha_2 + \alpha_0 \le (\tau_1 - \alpha_1)/\alpha_2. \tag{6.54c}$$

We proceed to check the validity of (6.52). When $k = 0$, and condition (6.54a), (6.54b) is satisfied, we multiply $||\mathbf{x}_0 - \mathbf{x}^*_0||$ on both sides of (6.54b), we have

$$||\mathbf{x}_0 - \mathbf{x}^*_0||^2 \le \frac{\tau_1 - \alpha_1}{\alpha_2}||\mathbf{x}_0 - \mathbf{x}^*_0||. \tag{6.55}$$

Multiplying both sides of (6.55) by α_2 and add α_0, get

$$\begin{aligned}\alpha_2\|\mathbf{x}_0-\mathbf{x}_0^*\|^2+\alpha_1\|\mathbf{x}_0-\mathbf{x}_0^*\|+\alpha_0 &\le \tau_1\|\mathbf{x}_0-\mathbf{x}_0^*\|+\alpha_0\\ &\le \tau_1(\tau_1-\alpha_1)/\alpha_2+\alpha_0.\end{aligned} \tag{6.56}$$

Combining (6.52) and (6.56), get

$$\|\mathbf{x}_1-\mathbf{x}_1^*\| \le \tau_1(\tau-\alpha_1)/\alpha_2+\alpha_0. \tag{6.57}$$

When condition (6.54c) is met, we have

$$\|\mathbf{x}_1-\mathbf{x}_1^*\| \le (\tau_1-\alpha_1)/\alpha_2. \tag{6.58}$$

Repeating the above process [cf.,(6.55)–(6.58)], we can derive

$$\|\mathbf{x}_k-\mathbf{x}_k^*\| \le (\tau_1-\alpha_1)/\alpha_2, \tag{6.59}$$

multiplying $\|\mathbf{x}_k-\mathbf{x}_k^*\|$ on both sides of (6.59) and adding α_0 , we can derive (6.53).

Therefore, when (6.54a)–(6.54c) is satisfied, the (6.53) is hold true for all k. The terms α_0 and α_1 are polynomial functions of sampling period h and the approximation levels K of the inverse of Hessian. Specially, when $h \to 0$ and $K \to \infty$, α_0, α_1 and α_2 can be made 0, $1-\gamma$ and $\frac{\gamma D_1}{2m_1}$, respectively. Formally

$$\lim_{h\to 0}\alpha_0=0, \quad \lim_{K\to\infty}\alpha_1=1-\gamma, \quad \lim_{h\to 0}\alpha_2=\frac{\gamma D_1}{2m_1}, \tag{6.60}$$

then conditions (6.54b) and (6.54c) hold with $\frac{\tau_1-\alpha_1}{\alpha_2}=\frac{2m_1(\tau_1+\gamma-1)}{D_1}$ and $\frac{\tau_1(\tau_1-\alpha_1)}{\alpha_2}=\frac{2m_1\tau_1(\tau_1+\gamma-1)}{D_1}$. According (6.52) and (6.53), the upper bound of $\|\mathbf{x}_{k+1}-\mathbf{x}_{k+1}^*\|$ can be rewritten as

$$\|\mathbf{x}_{k+1}-\mathbf{x}_{k+1}^*\| \le \tau_1\|\mathbf{x}_k-\mathbf{x}_k^*\|+\alpha_0. \tag{6.61}$$

By recursively applying (6.61) backward in time, we obtain

$$\|\mathbf{x}_k-\mathbf{x}_k^*\| \le \tau_1^k\|\mathbf{x}_0-\mathbf{x}_0^*\|+\alpha_0\left(\frac{1-\tau_1^k}{1-\tau_1}\right), \tag{6.62}$$

which $\tau_1 < 1$, after the expansion of the coefficients, we can derive

$$\|\mathbf{x}_k-\mathbf{x}_k^*\| \le \tau_1^k\|\mathbf{x}_0-\mathbf{x}_0^*\|+\left(\frac{1-\tau_1^k}{1-\tau_1}\right)\left(h^4\frac{D_1\Delta^2}{2m_1}+h^2\psi\Delta\right), \tag{6.63}$$

where $\psi=\frac{\gamma(L_1+M_1)\rho^{K+1}}{m_1}+1-\gamma$, when $K \to \infty$, we can derive (4.7).

Due to $0 < \rho < 1$, then when the approximate level K of the inverse Hessian matrix is chosen sufficiently large, i.e., $K \to \infty$, then the $\rho^K \to 0$, result $\psi = 0$, when $K \to \infty$, derive

$$\lim_{K\to\infty}\sup\|\mathbf{x}_{k+1}-\mathbf{x}^*(t_{k+1})\| = h^4\frac{D_1\Delta^2}{2m_1(1-\tau_1)}+h^2(1-\gamma)\Delta. \tag{6.64}$$

The proof is completed.

AIMS Press