A Fast Temporal Decomposition Procedure for Long-horizon Nonlinear Dynamic Programming

Sen Na
ICSI and Department of Statistics, University of California, Berkeley, senna@berkeley.edu

Mihai Anitescu
Mathematics and Computer Science Division, Argonne National Laboratory, anitescu@mcs.anl.gov

Mladen Kolar
Booth School of Business, University of Chicago, mladen.kolar@chicagobooth.edu

We propose a *fast* temporal decomposition procedure for solving long-horizon nonlinear dynamic programs. The core of the procedure is sequential quadratic programming (SQP) that utilizes a differentiable exact augmented Lagrangian as the merit function. Within each SQP iteration, we approximately solve the Newton system using an overlapping temporal decomposition strategy. We show that the approximate search direction is still a descent direction of the augmented Lagrangian, provided the overlap size and penalty parameters are suitably chosen, which allows us to establish the global convergence. Moreover, we show that a unit stepsize is accepted locally for the approximate search direction, and further establish a uniform, local linear convergence over stages. This local convergence rate matches the rate of the recent Schwarz scheme [38]. However, the Schwarz scheme has to solve nonlinear subproblems to optimality in each iteration, while we only perform a single Newton step instead. Numerical experiments validate our theories and demonstrate the superiority of our method.

Key words: nonlinear dynamic programming; temporal decomposition; sequential quadratic programming; augmented Lagrangian

1. Introduction

We consider nonlinear equality-constrained dynamic programs (NLDPs):

$$\min_{\boldsymbol{x},\boldsymbol{u}} \ \sum_{k=0}^{N-1} g_k(\boldsymbol{x}_k,\boldsymbol{u}_k)+g_N(\boldsymbol{x}_N), \tag{1.1a}$$

$$\text{s.t.} \ \ \boldsymbol{x}_{k+1}=f_k(\boldsymbol{x}_k,\boldsymbol{u}_k), \quad k=0,1,\ldots,N-1, \tag{1.1b}$$

$$\boldsymbol{x}_0=\bar{\boldsymbol{x}}_0, \tag{1.1c}$$

where $\boldsymbol{x}_k\in\mathbb{R}^{n_x}$ is the state variable, $\boldsymbol{u}_k\in\mathbb{R}^{n_u}$ is the control variable, $g_k:\mathbb{R}^{n_x}\times\mathbb{R}^{n_u}\to\mathbb{R}$ ($g_N:\mathbb{R}^{n_x}\to\mathbb{R}$) is the cost function, $f_k:\mathbb{R}^{n_x}\times\mathbb{R}^{n_u}\to\mathbb{R}^{n_x}$ is the dynamical constraint function, $\bar{\boldsymbol{x}}_0$ is the given initial state, and N is the temporal horizon length. In the control literature, Problem (1.1) is also called nonlinear optimal control problem. This paper focuses on solving (1.1) with a large N.

Large-scale nonlinear problems pose significant computational challenges due to the nonlinearity of the problems and the large number of variables that need to be optimized. Numerous solvers have been developed to address the scalability issue from different aspects. For example, IPOPT [60] is a primal-dual interior point method that employs the logarithmic barrier function with the filter line-search step. It enjoys both global and local superlinear convergence [58, 59]. As another example, Knitro [10] integrates the advantages of two complementary methods—the interior point method and the active-set method—in a unified package, and achieves the robust performance. We refer to [13, 43, 61] for the other scalable high-performance solvers for nonlinear programs.

The long-horizon NLDPs are of special interest as they appear in a variety of applications including portfolio management [56], autonomous vehicle [23, 68], and power planning [15]. The aforementioned centralized solvers, while exploiting the problem-dependent sparse structures to accelerate

the computation, are not particularly designed for NLDPs. As these solvers do not take advantage of generic properties of dynamical systems, they are deficient when directly implemented on Problem (1.1), especially when we have limited computing resources. The generic properties of dynamical systems, such as sensitivity and controllability, play a key role in developing various efficient algorithms [34, 52, 53, 54, 65]. Further, the centralized solvers are not flexible enough to be implemented on different types of computing hardware. Their performance heavily relies on a single high-speed processor, which sometimes is inaccessible. These limitations motivate us to design a parallel, DP-oriented procedure for (1.1).

In contrast to solving NLDPs in real time [16, 17, 18, 67], the dominant class of distributed offline methods is the decomposition-based methods, where the full problem is decomposed into multiple subproblems, which are then solved in a parallel environment [63]. For example, temporal decomposition [1, 30], Lagrangian dual decomposition [2, 31], and alternating direction method of multipliers (ADMM) [7, 42] are widely adopted in practice. The decomposition-based methods are preferable when the problem scale (i.e., N in our case) is so large that solving (1.1) on a single processor is computationally prohibitive, but multiple processors can be used to solve subproblems in parallel.

This paper contributes to the literature on the overlapping temporal decomposition (OTD) methods [1, 38, 52, 65]. A TD method decomposes the full temporal horizon $[0, N]$ into several short horizons $[0, N] \subseteq \cup_i [n_i, n_{i+1}]$; constructs subproblems $\{\mathcal{P}^i\}_i$ associated with short horizons $\{[n_i, n_{i+1}]\}_i$; solves all subproblems in parallel; and retrieves the full-horizon solution by composing the subproblem solutions sequentially. Different from the exclusive decomposition in TD, OTD extends each short horizon $[n_i, n_{i+1}]$ by b stages on two ends to encourage information exchange between two adjacent subproblems. The composition is performed by using only the exclusive part, $[n_i, n_{i+1}]$, of each subproblem's solution. OTD was first empirically studied in [1] on power planning problems, and rigorously analyzed for linear-quadratic convex DPs in [65]. In this work, the authors showed under the controllability and boundedness conditions that $\max_k \|(\hat{\boldsymbol{x}}_k - \boldsymbol{x}_k^\star; \hat{\boldsymbol{u}}_k - \boldsymbol{u}_k^\star)\| \leq C\rho^b$ for some constants $C > 0$ and $\rho \in (0, 1)$. Here, $(\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{u}}_k)$ is the OTD output at the stage k and $(\boldsymbol{x}_k^\star, \boldsymbol{u}_k^\star)$ is the optimal solution. For general NLDPs, [38, 52] recently proposed an overlapping Schwarz scheme as an application of OTD on nonlinear problems. We will review the Schwarz scheme in section 2 but briefly introduce it here to motivate our study.

Following the same spirit as OTD, the Schwarz scheme decomposes $[0, N]$ by $[0, N] \subseteq \cup_i [m_1^i, m_2^i]$, where $m_1^i = n_i - b$ and $m_2^i = n_{i+1} + b$ are two boundaries of the extended intervals. The i-th subproblem $\mathcal{P}^i = \mathcal{P}^i(\boldsymbol{d}_i)$ is parameterized by some boundary variables $\boldsymbol{d}_i = (\boldsymbol{d}_{m_1^i}; \boldsymbol{d}_{m_2^i})$. In the τ-th iteration, one first specifies the boundary variables $\boldsymbol{d}_i^\tau$ by the solutions of adjacent subproblems (noting that $m_1^i, m_2^i \notin [n_i, n_{i+1}]$); then solves the subproblems $\{\mathcal{P}^i(\boldsymbol{d}_i^\tau)\}_i$ to *optimality* in parallel and updates the boundary variables to $\boldsymbol{d}_i^{\tau+1}$. The solutions to subproblems are finally composed to derive a full-horizon solution. The Schwarz scheme is demonstrated in Figure 1. Clearly, the OTD step corresponds to one iteration of the Schwarz scheme. [38] empirically showed that the Schwarz scheme significantly improves the efficiency of ADMM; and may be as efficient as the centralized solver IPOPT, but provide significant flexibility on different computing environments. By adjusting the overlap size b of the decomposition, the scheme adapts to centralized or to decentralized environments. Furthermore, unlike ADMM whose convergence for nonlinear problems is established for some setups that may not directly apply to (1.1) (see [27, 62]), the Schwarz scheme exhibits a uniform local linear convergence. Under the same conditions of OTD on linear-quadratic convex DPs in [65], [38] showed $\max_k \|(\boldsymbol{x}_k^\tau - \boldsymbol{x}_k^\star; \boldsymbol{u}_k^\tau - \boldsymbol{u}_k^\star)\| \leq (C\rho^b)^\tau \max_k \|(\boldsymbol{x}_k^0 - \boldsymbol{x}_k^\star; \boldsymbol{u}_k^0 - \boldsymbol{u}_k^\star)\|$, where $(\boldsymbol{x}_k^\tau, \boldsymbol{u}_k^\tau)$ is the τ-th iterate of the Schwarz scheme. Despite this promising result, the Schwarz scheme has two limitations.

First, only local convergence guarantee is established while global convergence is still unknown. In fact, as we will explain later, the Schwarz scheme does not converge globally in general. There is no guarantee that the composed solution of the subproblems is the solution to the full problem, even if we correctly specify the boundary variables for each subproblem. Thus, we arise the question:

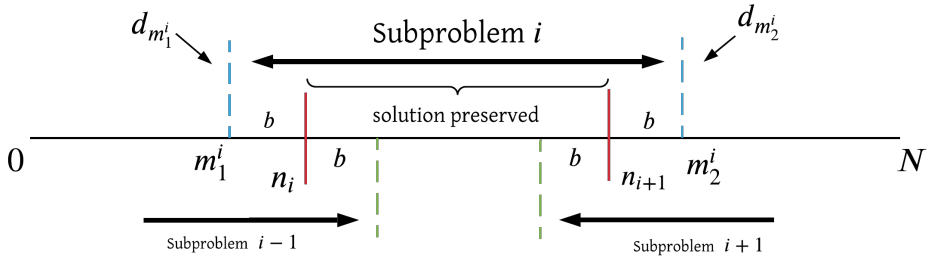


FIGURE 1. Demonstration of the Schwarz scheme. The horizontal line is the full horizon $[0, N]$. The red vertical lines are knots of the exclusive intervals; the blue vertical lines are boundaries of the extended intervals. The subproblem i parameterized by $\boldsymbol{d}_i = (\boldsymbol{d}_{m_1^i}, \boldsymbol{d}_{m_2^i})$ is solved, but only the exclusive part of the solution is used in the composition.

Q1: How to design an OTD-based procedure that converges globally?

Second, the Schwarz scheme is computationally expensive. In each iteration, the scheme solves all nonlinear subproblems $\{\mathcal{P}^i(\boldsymbol{d}_i)\}_i$ to *optimality*. However, we want to know:

Q2: Is it necessary to solve subproblems to optimality in each iteration to enjoy (uniform) local linear convergence?

We answer Q1 and Q2 by designing a *fast* OTD (FOTD) procedure. Our procedure is inspired by [63], where the author applied the sequential quadratic programming (SQP) to solve (1.1), and a parallel block-banded linear solver to solve the exact Newton system. By using the unit stepsize, [63] conducted a local analysis. In this paper, we propose a FOTD procedure to integrate global and local analyses. Similar to [63], FOTD is also built upon an SQP scheme. However, it utilizes an exact augmented Lagrangian merit function to adaptively select the stepsize via line search; and adopts an OTD strategy to approximately solve the Newton system. While there are many distributed methods for solving the Newton system exactly or approximately, e.g., [30, 39, 40], we specifically focus on OTD in order to build a relation to the Schwarz scheme. We prove that FOTD converges globally. A key technical step is to show that the approximate direction is a descent direction of the augmented Lagrangian, so that the iterates are improved towards the KKT point. This technical step justifies the choice of the augmented Lagrangian merit function. Furthermore, we show that the unit stepsize for the approximate direction is accepted locally, so that FOTD enjoys a uniform local linear convergence. Such a linear convergence matches the one of the Schwarz scheme [38]; however, FOTD requires much fewer computations as it does not solve nonlinear subproblems to optimality. Our experiments validate the theorems and demonstrate the superiority of FOTD.

Structure of the paper: In section 2, we present the preliminaries of OTD and review the Schwarz scheme. In section 3, we introduce our FOTD procedure. In section 4, we study the approximation error of the Newton system. The global and local convergence results are established in section 5 and section 6, respectively. Numerical experiments are presented in section 7 followed by conclusions in section 8. All the proofs are collected in appendices to make the main paper compact.

Notation: For an integer n, $[n] \coloneqq \{0, 1, \ldots, n\}$. For two integers n, m, we abuse the interval notation and let $[n, m]$, (n, m), $[n, m)$, $(n, m]$ be the corresponding index sets. All vectors in the paper are column vectors. For two vectors $\boldsymbol{a}$, $\boldsymbol{b}$, $(\boldsymbol{a}; \boldsymbol{b})$ denotes the column vector that stacks $\boldsymbol{a}$ and $\boldsymbol{b}$ sequentially. For a vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$, $\nabla f \in \mathbb{R}^{n \times m}$ is its Jacobian matrix. We let $\|\cdot\|$ denote the ℓ_2 norm for vectors and the spectral norm for matrices. For a symmetric matrix A, $\lambda_{\min}(A)$ denotes its smallest eigenvalue. We let I be the identity matrix and $\mathbf{0}$ be the zero matrix, whose dimensions can be inferred from the context. We also reserve the following notation. $\boldsymbol{x} = \boldsymbol{x}_{0:N} = (\boldsymbol{x}_0; \ldots; \boldsymbol{x}_N)$ is the state vector; $\boldsymbol{u} = \boldsymbol{u}_{0:N-1} = (\boldsymbol{u}_0; \ldots \boldsymbol{u}_{N-1})$ is the control vector; $\boldsymbol{z}_k = (\boldsymbol{x}_k; \boldsymbol{u}_k)$ $(\boldsymbol{z}_N = \boldsymbol{x}_N)$ is the state-control pair at stage k, and $\boldsymbol{z} = \boldsymbol{z}_{0:N} = (\boldsymbol{z}_0; \ldots; \boldsymbol{z}_N)$. We let $n_z = (N+1)n_x + N n_u$ and $\boldsymbol{z} \in \mathbb{R}^{n_z}$. We may also express $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{u})$ when explicitly specifying the components $\boldsymbol{x}, \boldsymbol{u}$ of $\boldsymbol{z}$.

2. Preliminaries We start by setting up an overlapping temporal decomposition (OTD). Given the full horizon $[0, N]$, we decompose it into M exclusive intervals with knots $0 = n_0 < n_1 < \cdots < n_M = N$. For example, we can choose evenly spaced knots $n_i = i \cdot N/M$ (suppose M is a divisor of N). Each one of these intervals is then extended by $b \geq 1$ stages on two ends; that is, the boundaries of the extended intervals are

$$m_1^i = \max\{n_i - b,\, 0\}, \qquad m_2^i = \min\{n_{i+1} + b,\, N\}, \qquad i = 0, 1, \ldots, M-1. \tag{2.1}$$

To simplify the notation, we use m_1, m_2 to denote the boundaries of a general extended interval i. We note that two successive extended intervals overlap on $2b$ stages.

For the extended interval i, we define the corresponding subproblem as

$$\mathcal{P}_\mu^i(\boldsymbol{d}_i): \quad \min_{\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{u}}_i} \sum_{k=m_1}^{m_2-1} g_k(\boldsymbol{x}_k, \boldsymbol{u}_k) + \tilde{g}_{m_2}(\boldsymbol{x}_{m_2}; \boldsymbol{d}_{i,2:4}), \tag{2.2a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{k+1} = f_k(\boldsymbol{x}_k, \boldsymbol{u}_k), \quad k \in [m_1, m_2), \tag{2.2b}$$

$$\boldsymbol{x}_{m_1} = \boldsymbol{d}_{i,1}, \tag{2.2c}$$

where $\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_{m_1:m_2}$ and $\tilde{\boldsymbol{u}}_i = \boldsymbol{u}_{m_1:m_2-1}$ are the state and control variables; $\boldsymbol{d}_i = \boldsymbol{d}_{i,1:4} = (\bar{\boldsymbol{x}}_{m_1}; \bar{\boldsymbol{x}}_{m_2}; \bar{\boldsymbol{u}}_{m_2}; \bar{\boldsymbol{\lambda}}_{m_2+1})$ are the boundary variables; $\tilde{g}_{m_2}(\boldsymbol{x}_{m_2}; \boldsymbol{d}_{i,2:4}) = g_N(\boldsymbol{x}_N)$ if $i = M-1$ (i.e., the last subproblem), otherwise for $\mu > 0$,

$$\tilde{g}_{m_2}(\boldsymbol{x}_{m_2}; \boldsymbol{d}_{i,2:4}) = g_{m_2}(\boldsymbol{x}_{m_2}, \bar{\boldsymbol{u}}_{m_2}) - \bar{\boldsymbol{\lambda}}_{m_2+1}^T f_{m_2}(\boldsymbol{x}_{m_2}, \bar{\boldsymbol{u}}_{m_2}) + \frac{\mu}{2}\|\boldsymbol{x}_{m_2} - \bar{\boldsymbol{x}}_{m_2}\|^2. \tag{2.3}$$

For the boundary variables $\boldsymbol{d}_i = (\boldsymbol{d}_{i,1}; \boldsymbol{d}_{i,2:4})$, $\boldsymbol{d}_{i,1} = \bar{\boldsymbol{x}}_{m_1}$ is the initial state variable; $\boldsymbol{d}_{i,2:4} = (\bar{\boldsymbol{x}}_{m_2}; \bar{\boldsymbol{u}}_{m_2}; \bar{\boldsymbol{\lambda}}_{m_2+1})$ are the terminal state, control, and dual variables. The boundary variables $\boldsymbol{d}_i = (\boldsymbol{d}_{i,1}; \boldsymbol{d}_{i,2:4})$ are given to the subproblem. In (2.3), μ is a *uniform* quadratic penalty parameter independent from i. In what follows, we denote by $\tilde{\boldsymbol{z}}_i = (\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{u}}_i)$ the ordered state-control vector of the subproblem i.

The subproblem $\mathcal{P}^i = \mathcal{P}_\mu^i(\boldsymbol{d}_i)$ in (2.2) is essentially the truncation of the full problem (1.1) onto the interval $[m_1, m_2]$, except that the initial state is fixed at $\boldsymbol{d}_{i,1} = \bar{\boldsymbol{x}}_{m_1}$ and the terminal cost on $\boldsymbol{x}_{m_2}$ is adjusted by $\tilde{g}_{m_2}(\boldsymbol{x}_{m_2}; \boldsymbol{d}_{i,2:4})$. The subproblem is parameterized by $\boldsymbol{d}_i$; we always specify $\boldsymbol{d}_i$ based on the previous iterate before solving $\mathcal{P}_\mu^i(\boldsymbol{d}_i)$. The formula (2.3) was first proposed by [35] for analyzing the real-time model predictive control schemes. The middle term depending on $\bar{\boldsymbol{\lambda}}_{m_2+1}$ and f_{m_2} reduces the KKT residual brought by the horizon truncation, and the quadratic penalty term *convexifies* the subproblem. The benefits of (2.3) will be clearer later. We clarify two corner cases: for $i = 0$ and $m_1 = 0$, $\bar{\boldsymbol{x}}_{m_1}$ is $\bar{\boldsymbol{x}}_0$ from Problem (1.1); for $i = M-1$ and $m_2 = N$, the adjustment on the terminal cost is restored. We mention that there exist other subproblem formulations in some restrictive setups: if g_{m_2}, f_{m_2} are separable, then $\bar{\boldsymbol{u}}_{m_2}$ is not needed in (2.3); if g_{m_2} is quadratic and convex and f_{m_2} is affine, then we can let $\mu = 0$. All these formulations ensure that the subproblem $\mathcal{P}_\mu^i(\boldsymbol{d}_i)$ is well-defined (e.g., is lower bounded) given a well-defined full problem.

Before introducing the Schwarz scheme, we need the following notation. We let $\boldsymbol{\lambda} = \boldsymbol{\lambda}_{0:N}$ be the dual variables of (1.1), where $\boldsymbol{\lambda}_0 \in \mathbb{R}^{n_x}$ is associated with the constraint in (1.1c) and $\boldsymbol{\lambda}_{k+1} \in \mathbb{R}^{n_x}$ for $k \in [N-1]$ is associated with the k-th constraint in (1.1b). Similarly, we let $\tilde{\boldsymbol{\lambda}}_i = \boldsymbol{\lambda}_{m_1:m_2}$ be the dual variables of (2.2). For the state variables $\tilde{\boldsymbol{x}}_i$ (similar for $\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{z}}_i, \tilde{\boldsymbol{\lambda}}_i$) and $k \in [m_1, m_2]$, $\tilde{\boldsymbol{x}}_{i,k}$ denotes the variable at the stage k in the subproblem i. For example, $k = n_i$ belongs to both subproblem $i-1$ and subproblem i; thus $\tilde{\boldsymbol{x}}_{i-1,n_i}$ and $\tilde{\boldsymbol{x}}_{i,n_i}$ refer to different variables at the same stage. The primal-dual solution of $\mathcal{P}_\mu^i(\boldsymbol{d}_i)$ is denoted by $(\tilde{\boldsymbol{z}}_i^\star(\boldsymbol{d}_i), \tilde{\boldsymbol{\lambda}}_i^\star(\boldsymbol{d}_i))$. We also define composition and decomposition operators in the next definition.

DEFINITION 2.1 **(composition and decomposition)**. For the subproblem variables $\{(\tilde{\boldsymbol{z}}_i, \tilde{\boldsymbol{\lambda}}_i)\}_{i=0}^{M-1}$, we define a composition operator $\mathcal{C}$ as $\mathcal{C}(\{(\tilde{\boldsymbol{z}}_i, \tilde{\boldsymbol{\lambda}}_i)\}_i) = (\boldsymbol{z}, \boldsymbol{\lambda})$, where $(\boldsymbol{z}_k, \boldsymbol{\lambda}_k) = (\tilde{\boldsymbol{z}}_{i,k}, \tilde{\boldsymbol{\lambda}}_{i,k})$ if $k \in$

Algorithm 1 Overlapping Schwarz Decomposition Procedure

Input: initial iterate $(\boldsymbol{z}^0, \boldsymbol{\lambda}^0)$ with $\boldsymbol{x}_0^0 = \bar{\boldsymbol{x}}_0$, a scalar $\mu > 0$;
for $\tau = 0, 1, 2, \ldots$ **do**
  for $i = 0, 1, \ldots, M-1$ **(in parallel) do**
    Let $\boldsymbol{d}_i^\tau = (\boldsymbol{x}_{m_1}^\tau; \boldsymbol{x}_{m_2}^\tau; \boldsymbol{u}_{m_2}^\tau; \boldsymbol{\lambda}_{m_2+1}^\tau)$ (note that $\boldsymbol{d}_{M-1}^\tau = \boldsymbol{x}_{m_1}^\tau$);
    Solve $\mathcal{P}_\mu^i(\boldsymbol{d}_i^\tau)$ to **optimality** and obtain the solution $(\tilde{\boldsymbol{z}}_i^\star(\boldsymbol{d}_i^\tau), \tilde{\boldsymbol{\lambda}}_i^\star(\boldsymbol{d}_i^\tau))$;
  end for
  Let $(\boldsymbol{z}^{\tau+1}, \boldsymbol{\lambda}^{\tau+1}) = \mathcal{C}(\{(\tilde{\boldsymbol{z}}_i^\star(\boldsymbol{d}_i^\tau), \tilde{\boldsymbol{\lambda}}_i^\star(\boldsymbol{d}_i^\tau))\}_i)$;
end for

$[n_i, n_{i+1})$ for $i \in [M-1]$, and $\boldsymbol{z}_N = \tilde{\boldsymbol{x}}_{M-1,N}$ and $\boldsymbol{\lambda}_N = \tilde{\boldsymbol{\lambda}}_{M-1,N}$. Conversely, for the full-horizon variable $(\boldsymbol{z}, \boldsymbol{\lambda})$, we define a decomposition operator $\mathcal{D}$ as $\mathcal{D}(\boldsymbol{z}, \boldsymbol{\lambda}) = \{\mathcal{D}_i(\boldsymbol{z}, \boldsymbol{\lambda})\}_{i=0}^{M-1}$, where $\mathcal{D}_i(\boldsymbol{z}, \boldsymbol{\lambda}) = (\tilde{\boldsymbol{z}}_i, \tilde{\boldsymbol{\lambda}}_i)$ with $\tilde{\boldsymbol{z}}_i = (\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{u}}_i) = (\boldsymbol{x}_{m_1:m_2}, \boldsymbol{u}_{m_1:m_2-1})$ and $\tilde{\boldsymbol{\lambda}}_i = \boldsymbol{\lambda}_{m_1:m_2}$.

From Definition 2.1, we see that the variables on the overlapping stages are discarded during the composition. That is, $\tilde{\boldsymbol{x}}_i = \tilde{\boldsymbol{x}}_{i,m_1:m_2}$ (similar for $\tilde{\boldsymbol{u}}_i, \tilde{\boldsymbol{\lambda}}_i$) only contributes $\tilde{\boldsymbol{x}}_{i,n_i:n_{i+1}-1}$ to the full vector.

Schwarz scheme. Given the τ-th iterate $(\boldsymbol{z}^\tau, \boldsymbol{\lambda}^\tau)$, we specify the boundary variables by $\boldsymbol{d}_i^\tau = (\boldsymbol{x}_{m_1}^\tau; \boldsymbol{x}_{m_2}^\tau; \boldsymbol{u}_{m_2}^\tau; \boldsymbol{\lambda}_{m_2+1}^\tau)$; solve the subproblems $\{\mathcal{P}_\mu^i(\boldsymbol{d}_i^\tau)\}_i$ in parallel to *optimality*; and obtain the solutions $\{(\tilde{\boldsymbol{z}}_i^\star(\boldsymbol{d}_i^\tau), \tilde{\boldsymbol{\lambda}}_i^\star(\boldsymbol{d}_i^\tau))\}_i$. Then, $(\boldsymbol{z}^{\tau+1}, \boldsymbol{\lambda}^{\tau+1}) = \mathcal{C}(\{(\tilde{\boldsymbol{z}}_i^\star(\boldsymbol{d}_i^\tau), \tilde{\boldsymbol{\lambda}}_i^\star(\boldsymbol{d}_i^\tau))\}_i)$. The Schwarz scheme is displayed in Algorithm 1. The convergence of the scheme is achieved by iteratively updating $\boldsymbol{d}_i$. One expects that, as τ increases, $\boldsymbol{d}_i^\tau$ becomes more precise so that the solution of $\mathcal{P}_\mu^i(\boldsymbol{d}_i^\tau)$ approaches to the truncated full solution. Thus, the composed solution of subproblems can recover the full solution.

The following result characterizes the relation between the optimality conditions of the subproblems (2.2) and the full problem (1.1).

THEOREM 2.1 **(relation of KKT conditions)**. *For any scalar μ, we have two cases:*
(i) *Suppose $(\boldsymbol{z}^\star, \boldsymbol{\lambda}^\star)$ is a KKT point of (1.1), then $\mathcal{D}_i(\boldsymbol{z}^\star, \boldsymbol{\lambda}^\star)$, $\forall i \in [M-1]$, is a KKT point of $\mathcal{P}_\mu^i(\boldsymbol{d}_i^\star)$ where $\boldsymbol{d}_i^\star = (\boldsymbol{x}_{m_1}^\star; \boldsymbol{x}_{m_2}^\star; \boldsymbol{u}_{m_2}^\star; \boldsymbol{\lambda}_{m_2+1}^\star)$;*
(ii) *Suppose $(\tilde{\boldsymbol{z}}_i^\star, \tilde{\boldsymbol{\lambda}}_i^\star) = (\tilde{\boldsymbol{z}}_i^\star(\boldsymbol{d}_i), \tilde{\boldsymbol{\lambda}}_i^\star(\boldsymbol{d}_i))$, $\forall i \in [M-1]$, is a KKT point of $\mathcal{P}_\mu^i(\boldsymbol{d}_i)$ with any boundary variables $\boldsymbol{d}_i$ satisfying $\boldsymbol{d}_{0,1} = \bar{\boldsymbol{x}}_0$, then $\mathcal{C}(\{(\tilde{\boldsymbol{z}}_i^\star, \tilde{\boldsymbol{\lambda}}_i^\star)\}_i)$ is a KKT point of (1.1) if and only if the solutions of any two successive subproblems are compatible at the common boundaries. That is, for any $i \in [1, M-1]$ and knot $k = n_i$, we have*

$$\tilde{\boldsymbol{x}}_{i,k}^\star = \tilde{\boldsymbol{x}}_{i-1,k}^\star, \quad A_{k-1}^T(\tilde{\boldsymbol{\lambda}}_{i,k}^\star - \tilde{\boldsymbol{\lambda}}_{i-1,k}^\star) = \boldsymbol{0}, \quad B_{k-1}^T(\tilde{\boldsymbol{\lambda}}_{i,k}^\star - \tilde{\boldsymbol{\lambda}}_{i-1,k}^\star) = \boldsymbol{0}, \tag{2.4}$$

where $A_{k-1} = \nabla_{\boldsymbol{x}_{k-1}}^T f_{k-1} \in \mathbb{R}^{n_x \times n_x}$, $B_{k-1} = \nabla_{\boldsymbol{u}_{k-1}}^T f_{k-1} \in \mathbb{R}^{n_x \times n_u}$ are the Jacobian matrices of f_{k-1} evaluated at $\tilde{\boldsymbol{z}}_{i-1,k-1}^\star$.

Proof. See Appendix A.1. □

From Theorem 2.1(i), we can see that the truncated KKT point $\mathcal{D}_i(\boldsymbol{z}^\star, \boldsymbol{\lambda}^\star)$ is also a KKT point of the subproblem, provided the boundary variables are correctly specified (i.e., $\boldsymbol{d}_i = \boldsymbol{d}_i^\star$). However, Theorem 2.1(ii) provides a negative view: even if one obtains a KKT point for each subproblem with any boundary variables, composing the solutions together does not necessarily result in a KKT point of the full problem. Only if the successive KKT points are compatible on knots $\{n_i\}_{i=1}^{M-1}$ can we guarantee to have a full-horizon KKT point. The subtlety lies in the fact that the solution at stage n_i is from subproblem i, while the solution at stage $n_i - 1$ is from subproblem $i-1$. Thus, (2.4) is needed to link solutions from two successive subproblems. By Theorem 2.1(ii), we know the Schwarz scheme, which simply composes subproblem solutions, may not converge globally in general.

However, if $(\boldsymbol{z}^0, \boldsymbol{\lambda}^0)$ is sufficiently close to $(\boldsymbol{z}^\star, \boldsymbol{\lambda}^\star)$, [38] showed that for $i \in [M-1]$, $\mathcal{P}_\mu^i(\boldsymbol{d}_i)$ has a unique solution $\mathcal{D}_i(\boldsymbol{z}^\star, \boldsymbol{\lambda}^\star)$ in a neighborhood of $\boldsymbol{d}_i^\star$. Thus, one expects $(\tilde{\boldsymbol{z}}_i^\star(\boldsymbol{d}_i^\tau), \tilde{\boldsymbol{\lambda}}_i^\star(\boldsymbol{d}_i^\tau)) \to \mathcal{D}_i(\boldsymbol{z}^\star, \boldsymbol{\lambda}^\star)$ as $\tau \to \infty$, and $\mathcal{C}(\{(\tilde{\boldsymbol{z}}_i^\star(\boldsymbol{d}_i^\tau), \tilde{\boldsymbol{\lambda}}_i^\star(\boldsymbol{d}_i^\tau))\}_i) \to \mathcal{C}(\{\mathcal{D}_i(\boldsymbol{z}^\star, \boldsymbol{\lambda}^\star)\}_i) = (\boldsymbol{z}^\star, \boldsymbol{\lambda}^\star)$. Then, the local convergence of the Schwarz scheme is ensured. Specifically, [38] proved for some $C > 0$ and $\rho \in (0,1)$ that

$$\max_k \|(\boldsymbol{z}_k^\tau - \boldsymbol{z}_k^\star; \boldsymbol{\lambda}_k^\tau - \boldsymbol{\lambda}_k^\star)\| \le (C\rho^b)^\tau \cdot \max_k \|(\boldsymbol{z}_k^0 - \boldsymbol{z}_k^\star; \boldsymbol{\lambda}_k^0 - \boldsymbol{\lambda}_k^\star)\|. \tag{2.5}$$

In addition to lacking global convergence, the Schwarz scheme is computationally intensive, since Line 5 of Algorithm 1 requires finding optimal solutions to nonlinear subproblems. In the next section, we relax this computational requirement by using an SQP scheme, and finally show in section 6 that (2.5) holds even when Line 5 is substituted by one Newton step (cf. Theorem 6.3).

3. Embedding OTD into SQP We note that the lack of global convergence of the Schwarz scheme is due to the lack of a coordinator, which can monitor the convergence progress towards the full solution. This motivates us to solve (1.1) under an SQP framework. For each SQP iteration, we apply OTD to approximately solve the Newton system, which corresponds to a linear-quadratic DP problem. Although other distributed methods are also applicable, we are particularly interested in OTD since it reveals a nice relation to the Schwarz scheme. By embedding OTD into SQP, we are able to establish global convergence, which resolves one of the limitations of the Schwarz scheme.

We write Problem (1.1) in a compact form by

$$\min_{\boldsymbol{z}} \; g(\boldsymbol{z}), \qquad \text{s.t. } f(\boldsymbol{z}) = \boldsymbol{0},$$

where

$$\begin{aligned} g(\boldsymbol{z}) &= \sum_{k=0}^{N} g_k(\boldsymbol{z}_k) = \sum_{k=0}^{N-1} g_k(\boldsymbol{x}_k, \boldsymbol{u}_k) + g_N(\boldsymbol{x}_N), \\ f(\boldsymbol{z}) &= (\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0; \boldsymbol{x}_1 - f_0(\boldsymbol{z}_0); \ldots; \boldsymbol{x}_N - f_{N-1}(\boldsymbol{z}_{N-1})) \\ &= (\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0; \boldsymbol{x}_1 - f_0(\boldsymbol{x}_0, \boldsymbol{u}_0); \ldots; \boldsymbol{x}_N - f_{N-1}(\boldsymbol{x}_{N-1}, \boldsymbol{u}_{N-1})). \end{aligned} \tag{3.1}$$

The Lagrangian function is $\mathcal{L}(\boldsymbol{z}, \boldsymbol{\lambda}) = g(\boldsymbol{z}) + \boldsymbol{\lambda}^T f(\boldsymbol{z})$ and the KKT conditions are $\nabla_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}^\star, \boldsymbol{\lambda}^\star) = \boldsymbol{0}$, $\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{z}^\star, \boldsymbol{\lambda}^\star) = \boldsymbol{0}$. The SQP scheme applies Newton's method on the KKT system. In particular, given the τ-th iterate $(\boldsymbol{z}^\tau, \boldsymbol{\lambda}^\tau)$, the Newton direction $(\Delta \boldsymbol{z}^\tau, \Delta \boldsymbol{\lambda}^\tau)$ is obtained by solving

$$\begin{pmatrix} \hat{H}^\tau & (G^\tau)^T \\ G^\tau & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \Delta \boldsymbol{z}^\tau \\ \Delta \boldsymbol{\lambda}^\tau \end{pmatrix} = - \begin{pmatrix} \nabla_{\boldsymbol{z}} \mathcal{L}^\tau \\ \nabla_{\boldsymbol{\lambda}} \mathcal{L}^\tau \end{pmatrix}, \tag{3.2}$$

where $\nabla_{\boldsymbol{z}} \mathcal{L}^\tau = \nabla_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}^\tau, \boldsymbol{\lambda}^\tau)$, $\nabla_{\boldsymbol{\lambda}} \mathcal{L}^\tau = \nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{z}^\tau, \boldsymbol{\lambda}^\tau)$, $G^\tau = \nabla_{\boldsymbol{z}}^T f(\boldsymbol{z}^\tau)$, and $\hat{H}^\tau$ is a modification of the Hessian $H^\tau = \nabla_{\boldsymbol{z}}^2 \mathcal{L}(\boldsymbol{z}^\tau, \boldsymbol{\lambda}^\tau)$ that preserves the block diagonal structure of H^τ. The goal of the modification is to let $\hat{H}^\tau$ be positive definite in the null space $\{\boldsymbol{z} : G^\tau \boldsymbol{z} = \boldsymbol{0}\}$, if H^τ is not. For example, a simple structure-preserving modification is the Levenberg-style modification [20]: $\hat{H}^\tau = H^\tau + \gamma I$ for a suitably large $\gamma > 0$. Other Hessian modification methods are referred to in [41]. The explicit formula of the system (3.2) is displayed in Problem (3.8).

Given the Newton direction $(\Delta \boldsymbol{z}^\tau, \Delta \boldsymbol{\lambda}^\tau)$ from (3.2), the SQP iterate is updated as

$$\begin{pmatrix} \boldsymbol{z}^{\tau+1} \\ \boldsymbol{\lambda}^{\tau+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{z}^\tau \\ \boldsymbol{\lambda}^\tau \end{pmatrix} + \alpha_\tau \begin{pmatrix} \Delta \boldsymbol{z}^\tau \\ \Delta \boldsymbol{\lambda}^\tau \end{pmatrix}, \tag{3.3}$$

with the stepsize α_τ being selected by passing a line search condition based on a merit function. We employ the following differentiable exact augmented Lagrangian as the merit function

$$\mathcal{L}_\eta(\boldsymbol{z}, \boldsymbol{\lambda}) = \mathcal{L}(\boldsymbol{z}, \boldsymbol{\lambda}) + \frac{\eta_1}{2} \|\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{z}, \boldsymbol{\lambda})\|^2 + \frac{\eta_2}{2} \|\nabla_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z}, \boldsymbol{\lambda})\|^2, \tag{3.4}$$

where $\eta = (\eta_1, \eta_2)$ are the penalty parameters. The first penalty biases the feasibility error, while the second penalty biases the optimality error. The function (3.4) is called exact augmented Lagrangian, since one can show that the solution of the unconstrained problem $\min_{\boldsymbol{z},\boldsymbol{\lambda}} \mathcal{L}_\eta(\boldsymbol{z},\boldsymbol{\lambda})$ is also the solution of (1.1), provided η_1 is large enough and η_2 is small enough [3, Proposition 4.15]. We refer to [36, 37, 45, 46, 69] for more studies of (3.4) on constrained optimization problems. We should mention that there are many penalty functions that can be used as a merit function; however, (3.4) is particularly suitable and important for our analysis. First, recalling the subproblem terminal cost (2.3), we need an approximation of the terminal dual variable, which means that the merit function has to endure a dual perturbation. Such a requirement rules out the penalty functions that depend only on the primal variables $\boldsymbol{z}$. Second, for the SQP schemes, the differentiable merit functions such as (3.4) can overcome the Maratos effect and locally accept a unit stepsize, which is critical to have a fast local convergence rate. In contrast, the non-smooth merit functions suffer from the Maratos effect and require non-trivial modifications (e.g., the second-order correction) of the SQP schemes to achieve a fast local rate [41, Chapter 15.5]. We will clearly see the benefits of the merit function (3.4) later. In particular, see the discussion below Theorem 5.1 and see Theorem 6.1 as well.

With (3.4), the stepsize α_τ is selected to make the following Armijo condition hold

$$\mathcal{L}_\eta^{\tau+1} \le \mathcal{L}_\eta^\tau + \beta\alpha_\tau \begin{pmatrix} \nabla_{\boldsymbol{z}} \mathcal{L}_\eta^\tau \\ \nabla_{\boldsymbol{\lambda}} \mathcal{L}_\eta^\tau \end{pmatrix}^T \begin{pmatrix} \Delta \boldsymbol{z}^\tau \\ \Delta \boldsymbol{\lambda}^\tau \end{pmatrix}, \tag{3.5}$$

where $\beta \in (0, 1/2)$ is a prespecified parameter, $\mathcal{L}_\eta^\tau = \mathcal{L}_\eta(\boldsymbol{z}^\tau, \boldsymbol{\lambda}^\tau)$ (similar for $\mathcal{L}_\eta^{\tau+1}$, $\nabla\mathcal{L}_\eta^\tau$), and

$$\begin{pmatrix} \nabla_{\boldsymbol{z}} \mathcal{L}_\eta(\boldsymbol{z},\boldsymbol{\lambda}) \\ \nabla_{\boldsymbol{\lambda}} \mathcal{L}_\eta(\boldsymbol{z},\boldsymbol{\lambda}) \end{pmatrix} = \begin{pmatrix} I + \eta_2 H(\boldsymbol{z},\boldsymbol{\lambda}) & \eta_1 G^T(\boldsymbol{z}) \\ \eta_2 G(\boldsymbol{z}) & I \end{pmatrix} \begin{pmatrix} \nabla_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{z},\boldsymbol{\lambda}) \\ \nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{z},\boldsymbol{\lambda}) \end{pmatrix}. \tag{3.6}$$

Among the steps in (3.2), (3.3), and (3.5), solving the Newton system in (3.2) is often the most computationally expensive step. Thus, we apply the decomposition method, OTD, to solve (3.2) approximately. We obtain an approximate direction $(\tilde{\Delta}\boldsymbol{z}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau)$, and then use it in the steps of (3.3) and (3.5). We introduce some extra notation. We let $H(\boldsymbol{z},\boldsymbol{\lambda}) = \nabla_{\boldsymbol{z}}^2 \mathcal{L}(\boldsymbol{z},\boldsymbol{\lambda}) = \operatorname{diag}(H_0, \dots, H_N)$ be the Hessian of $\mathcal{L}(\boldsymbol{z},\boldsymbol{\lambda})$ with respect to $\boldsymbol{z}$, where

$$H_k(\boldsymbol{z}_k, \boldsymbol{\lambda}_{k+1}) = \begin{pmatrix} Q_k & S_k^T \\ S_k & R_k \end{pmatrix} = \begin{pmatrix} \nabla_{\boldsymbol{x}_k}^2 \mathcal{L} & \nabla_{\boldsymbol{x}_k \boldsymbol{u}_k} \mathcal{L} \\ \nabla_{\boldsymbol{u}_k \boldsymbol{x}_k} \mathcal{L} & \nabla_{\boldsymbol{u}_k}^2 \mathcal{L} \end{pmatrix}, \quad \forall k \in [N-1], \quad H_N(\boldsymbol{z}_N) = \nabla_{\boldsymbol{x}_N}^2 \mathcal{L}. \tag{3.7}$$

Note that H does not depend on $\boldsymbol{\lambda}_0$. We also let $A_k(\boldsymbol{z}_k) = \nabla_{\boldsymbol{x}_k}^T f_k(\boldsymbol{z}_k)$ and $B_k(\boldsymbol{z}_k) = \nabla_{\boldsymbol{u}_k}^T f_k(\boldsymbol{z}_k)$. To simplify the notation, we suppress the evaluation points in $\{H_k, A_k, B_k\}$ and suppress the iteration index τ to refer to a general τ-th iteration of (3.2). With the same block-diagonal structure as H, we have $\hat{H} = \operatorname{diag}(\hat{H}_0, \dots, \hat{H}_N)$, and use $\hat{Q}_k, \hat{S}_k, \hat{R}_k$ to denote the components of $\hat{H}_k$.

Solving (3.2) is equivalent to solving the following linear-quadratic DP problem:

$$\min_{\boldsymbol{p},\boldsymbol{q}} \ \sum_{k=0}^{N-1} \left\{ \frac{1}{2} \begin{pmatrix} \boldsymbol{p}_k \\ \boldsymbol{q}_k \end{pmatrix}^T \begin{pmatrix} \hat{Q}_k & \hat{S}_k^T \\ \hat{S}_k & \hat{R}_k \end{pmatrix} \begin{pmatrix} \boldsymbol{p}_k \\ \boldsymbol{q}_k \end{pmatrix} + \begin{pmatrix} \nabla_{\boldsymbol{x}_k} \mathcal{L} \\ \nabla_{\boldsymbol{u}_k} \mathcal{L} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{p}_k \\ \boldsymbol{q}_k \end{pmatrix} \right\} + \frac{1}{2} \boldsymbol{p}_N^T \hat{Q}_N \boldsymbol{p}_N + \nabla_{\boldsymbol{x}_N}^T \mathcal{L} \boldsymbol{p}_N, \tag{3.8a}$$

$$\text{s.t.} \ \ \boldsymbol{p}_{k+1} = A_k \boldsymbol{p}_k + B_k \boldsymbol{q}_k - \nabla_{\boldsymbol{\lambda}_{k+1}} \mathcal{L}, \quad k \in [N-1], \tag{3.8b}$$

$$\boldsymbol{p}_0 = -\nabla_{\boldsymbol{\lambda}_0} \mathcal{L}. \tag{3.8c}$$

We let $\boldsymbol{\zeta} = \boldsymbol{\zeta}_{0:N}$ be the dual variables and $\boldsymbol{w} = \boldsymbol{w}_{0:N} = (\boldsymbol{p}, \boldsymbol{q})$ be the primal variables of Problem (3.8). By the recursive constraints in (3.8b)-(3.8c), we know that the Jacobian G (in any iteration) has a "staircase" structure: for all $k \in [N]$, the $(k, 2k-1)$-block matrix is an identity matrix. Thus, G has full row rank. Suppose $\hat{H}$ is positive definite in $\{\boldsymbol{z} : G\boldsymbol{z} = \boldsymbol{0}\}$, then the KKT matrix in (3.2) is nonsingular; (3.8) has a unique global solution $(\boldsymbol{w}^\star, \boldsymbol{\zeta}^\star)$; and $(\Delta\boldsymbol{z}, \Delta\boldsymbol{\lambda}) = (\boldsymbol{w}^\star, \boldsymbol{\zeta}^\star)$ [41, Theorem 16.2]. Note that the linear terms in (3.8a) are given by the gradients of the Lagrangian. With this formulation, our dual solution $\boldsymbol{\zeta}^\star$ of (3.8) is the dual direction $\Delta\boldsymbol{\lambda}$. When the linear terms in (3.8a) are given by the gradients of the objective g, the dual solution $\boldsymbol{\zeta}^\star$ would be the dual iterate $\boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}$.

Algorithm 2 A Fast Overlapping Temporal Decomposition Procedure

1: **Input:** initial iterate $(\boldsymbol{z}^0, \boldsymbol{\lambda}^0)$ with $\boldsymbol{x}_0^0 = \bar{\boldsymbol{x}}_0$, scalars $\mu, \eta_1, \eta_2 > 0$, $\beta \in (0, 1/2)$;
2: **for** $\tau = 0, 1, 2, \ldots$ **do**
3: Compute $\hat{H}^\tau$, G^τ, and $\nabla \mathcal{L}^\tau$;
4: **for** $i = 0, 1, \ldots, M-1$ **(in parallel) do**
5: Let $\boldsymbol{d}_i^\tau = (\mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0})$ (note that $\boldsymbol{d}_{M-1}^\tau = \mathbf{0}$);
6: Solve $\mathcal{LP}_\mu^i(\boldsymbol{d}_i^\tau)$ to **optimality** and obtain the solution $(\tilde{\boldsymbol{w}}_i^\star(\boldsymbol{d}_i^\tau), \tilde{\boldsymbol{\zeta}}_i^\star(\boldsymbol{d}_i^\tau))$;
7: **end for**
8: Let $(\tilde{\Delta}\boldsymbol{z}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau) = \mathcal{C}(\{(\tilde{\boldsymbol{w}}_i^\star(\boldsymbol{d}_i^\tau), \tilde{\boldsymbol{\zeta}}_i^\star(\boldsymbol{d}_i^\tau))\}_i)$;
9: Select α_τ by a line search based on (3.5) and update the iterate by (3.3);
10: **end for**

We now apply OTD on (3.8). By an analogy to (2.2), the subproblem i, $\mathcal{LP}_\mu^i(\boldsymbol{d}_i)$, is defined as

$$\min_{\tilde{\boldsymbol{p}}_i, \tilde{\boldsymbol{q}}_i} \sum_{k=m_1}^{m_2-1} \left\{ \frac{1}{2} \begin{pmatrix} \boldsymbol{p}_k \\ \boldsymbol{q}_k \end{pmatrix}^T \begin{pmatrix} \hat{Q}_k & \hat{S}_k^T \\ \hat{S}_k & \hat{R}_k \end{pmatrix} \begin{pmatrix} \boldsymbol{p}_k \\ \boldsymbol{q}_k \end{pmatrix} + \begin{pmatrix} \nabla_{\boldsymbol{x}_k} \mathcal{L} \\ \nabla_{\boldsymbol{u}_k} \mathcal{L} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{p}_k \\ \boldsymbol{q}_k \end{pmatrix} \right\} \tag{3.9a}$$

$$+ \frac{1}{2} \begin{pmatrix} \boldsymbol{p}_{m_2} \\ \bar{\boldsymbol{q}}_{m_2} \end{pmatrix}^T \begin{pmatrix} \hat{Q}_{m_2} & \hat{S}_{m_2}^T \\ \hat{S}_{m_2} & \hat{R}_{m_2} \end{pmatrix} \begin{pmatrix} \boldsymbol{p}_{m_2} \\ \bar{\boldsymbol{q}}_{m_2} \end{pmatrix} + \boldsymbol{p}_{m_2}^T (\nabla_{\boldsymbol{x}_{m_2}} \mathcal{L} - A_{m_2}^T \bar{\boldsymbol{\zeta}}_{m_2+1}) + \frac{\mu}{2} \|\boldsymbol{p}_{m_2} - \bar{\boldsymbol{p}}_{m_2}\|^2,$$

$$\text{s.t. } \boldsymbol{p}_{k+1} = A_k \boldsymbol{p}_k + B_k \boldsymbol{q}_k - \nabla_{\boldsymbol{\lambda}_{k+1}} \mathcal{L}, \quad k \in [m_1, m_2), \tag{3.9b}$$

$$\boldsymbol{p}_{m_1} = \bar{\boldsymbol{p}}_{m_1}. \tag{3.9c}$$

With slight abuse of notation, $\boldsymbol{d}_i = \boldsymbol{d}_{i,1:4} = (\bar{\boldsymbol{p}}_{m_1}; \bar{\boldsymbol{p}}_{m_2}; \bar{\boldsymbol{q}}_{m_2}; \bar{\boldsymbol{\zeta}}_{m_2+1})$ are the boundary variables; $\tilde{\boldsymbol{w}}_i = (\tilde{\boldsymbol{p}}_i, \tilde{\boldsymbol{q}}_i)$, $\tilde{\boldsymbol{\zeta}}_i = \boldsymbol{\zeta}_{m_1:m_2}$ are the primal, dual variables of $\mathcal{LP}_\mu^i(\boldsymbol{d}_i)$; and $(\tilde{\boldsymbol{w}}_i^\star(\boldsymbol{d}_i), \tilde{\boldsymbol{\zeta}}_i^\star(\boldsymbol{d}_i))$ is the solution.

FOTD scheme. We set the stage to present the FOTD procedure. FOTD consists of three steps: given the τ-th iterate $(\boldsymbol{z}^\tau, \boldsymbol{\lambda}^\tau)$,

Step 1: Compute Hessian $\hat{H}^\tau$, Jacobian G^τ, and KKT residual vector $\nabla \mathcal{L}^\tau$.

Step 2: Solve subproblems $\{\mathcal{LP}_\mu^i(\boldsymbol{d}_i^\tau)\}_i$ with $\boldsymbol{d}_i^\tau = (\mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0})$ and obtain solutions $\{(\tilde{\boldsymbol{w}}_i^\star(\boldsymbol{d}_i^\tau), \tilde{\boldsymbol{\zeta}}_i^\star(\boldsymbol{d}_i^\tau))\}_i$. Then, $(\tilde{\Delta}\boldsymbol{z}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau) = \mathcal{C}(\{(\tilde{\boldsymbol{w}}_i^\star(\boldsymbol{d}_i^\tau), \tilde{\boldsymbol{\zeta}}_i^\star(\boldsymbol{d}_i^\tau))\}_i)$.

Step 3: Update the iterate by (3.3) with α_τ being selected by a line search based on (3.5).

We summarize FOTD in Algorithm 2 and present several remarks.

REMARK 3.1 **(necessity of the quadratic penalty)**. Even if the full problem (3.8) has a unique global solution, this does not necessarily imply that the subproblem (3.9) has a unique solution. Consider the following example. Let $N = 2$, $n_x = n_u = 1$, $\hat{Q}_0 = \hat{R}_0 = 1$, $\hat{Q}_1 = -\hat{R}_1 = -2$, $\hat{Q}_2 = 3$, $\hat{S}_0 = \hat{S}_1 = 0$, $A_0 = A_1 = B_0 = B_1 = 1$, and all linear terms in (3.8a) are zero. Then, the full problem: $\min_{\boldsymbol{p},\boldsymbol{q}} p_0^2 + q_0^2 - 2p_1^2 + 2q_1^2 + 3p_2^2$, s.t. $p_0 = 0$, $p_1 = p_0 + q_0$, $p_2 = p_1 + q_1$, has a unique global solution $(\boldsymbol{p}^\star, \boldsymbol{q}^\star) = (\mathbf{0}, \mathbf{0})$. However, when we truncate onto $[0, 1]$, the subproblem has a quadratic objective with the square matrix $\mathrm{diag}(1, 1, -2+\mu)$ and constraints $p_0 = 0$, $p_1 = p_0 + q_0$. Thus, by plugging the constraints into the objective, we can easily obtain that, when $\mu < 1$, the subproblem is unbounded below; when $\mu = 1$, the subproblem has infinitely many solutions; when $\mu > 1$, the subproblem has a unique global solution. Thus, to ensure that the subproblem has a unique global solution, a quadratic penalty with large enough μ on the terminal stage is necessary (cf. (3.9a)).

REMARK 3.2. We can express the optimality conditions of $\mathcal{LP}_\mu^i(\boldsymbol{d}_i)$ using a Newton system like (3.2). We observe that the KKT matrix depends on $\{A_k, B_k, \hat{H}_k\}_{k=m_1}^{m_2-1} \cup \{\hat{Q}_{m_2} + \mu I\}$, but not on $\boldsymbol{d}_i$. Thus, the uniqueness of the solution of $\mathcal{LP}_\mu^i(\boldsymbol{d}_i)$ is independent from $\boldsymbol{d}_i$. By Theorem 2.1(i), if $\boldsymbol{d}_i^\star = (\Delta\boldsymbol{x}_{m_1}; \Delta\boldsymbol{z}_{m_2}; \Delta\boldsymbol{\lambda}_{m_2+1})$, then $(\tilde{\boldsymbol{w}}_i^\star(\boldsymbol{d}_i^\star), \tilde{\boldsymbol{\zeta}}_i^\star(\boldsymbol{d}_i^\star)) = (\Delta\boldsymbol{x}_{m_1:m_2}, \Delta\boldsymbol{u}_{m_1:m_2-1}, \Delta\boldsymbol{\lambda}_{m_1:m_2})$. However, since there is no good guess for a search direction, we always set $\boldsymbol{d}_i = (\mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0})$ in Algorithm 2 (Line 5).

REMARK 3.3. Comparing Line 6 of Algorithm 2 with Line 5 of Algorithm 1, FOTD also solves subproblems $\{\mathcal{LP}_\mu^i\}_i$ to optimality. However, $\mathcal{LP}_\mu^i$ is a linear-quadratic DP, which can be solved efficiently while solving $\mathcal{P}_\mu^i$ is more expensive. The problem $\mathcal{LP}_\mu^i$ can be regarded as a linearization of the problem $\mathcal{P}_\mu^i$; thus solving $\mathcal{LP}_\mu^i$ corresponds to computing one Newton step of $\mathcal{P}_\mu^i$.

REMARK 3.4. Two SQP components, Hessian modification and line search, are less addressed in this paper. Performing them efficiently in a parallel environment is of course desirable from a practical aspect, while we put the implementation details of this regard aside in the paper but briefly discuss in this remark. Our paper is mainly concerned with solving the Newton system (3.2), which always requires more computations.

For line search, we note that the augmented Lagrangian $\mathcal{L}_\eta$ and its gradient $\nabla\mathcal{L}_\eta$ are separable in stages. Thus, we can easily evaluate them in parallel, where each processor computes a short horizon e.g., $[n_i, n_{i+1})$, and a coordinator is needed to sum up the results of all processors to check (3.5).

For Hessian modification, a desirable modification $\hat{H}$ should be close to the Hessian H whenever H satisfies the second-order sufficient condition (SOSC, i.e., (4.1) with $\hat{H}$ being replaced by H). If H does not satisfy the condition, we regularize H, e.g. $\hat{H} = H + (\gamma_{RH} + \|H\|)I$, to let $\hat{H}$ satisfy the condition, which ensures that the full Newton system (3.2) has a unique solution. See Assumptions 4.1 and 6.1. Thus, we have to check the positive definiteness of the reduced Hessian $Z^T H Z$, where the columns of Z span the null space of the Jacobian $G = \nabla_{\boldsymbol{z}}^T f$. It is equivalent to testing for positive definiteness of $H + c \cdot G^T G$ with a scalar c sufficiently large [41]. Note that $H + c \cdot G^T G$ is a block-tridiagonal matrix; thus, we can apply the parallel Cholesky factorization to check its definiteness in a parallel environment [12, 63, 64]. As analyzed in [12, Section 2.2], the total flops of the parallel Cholesky of M processors are less than $7N(n_x + n_u)^3$, so the average flop of a single processor is in order of $O(N(n_x + n_u)^3/M)$, which does not grow with N if the ratio N/M is a constant.

REMARK 3.5 **(extensions to inequality constraints)**. The FOTD procedure can be generalized to inequality-constrained problems. In particular, with inequality constraints $h_k(\boldsymbol{x}_k, \boldsymbol{u}_k) \le \boldsymbol{0}$, $k \in [N-1]$, the full SQP problem (3.8) will additionally have the linearized inequality constraints $C_k \boldsymbol{p}_k + D_k \boldsymbol{q}_k \le \boldsymbol{0}$, $k \in [N-1]$, where $C_k = \nabla_{\boldsymbol{x}_k}^T h_k$ and $D_k = \nabla_{\boldsymbol{u}_k}^T h_k$ are the Jacobian matrices of h_k. The FOTD scheme still applies OTD on the full problem (3.8) so that the subproblems (3.9) are inequality-constrained quadratic programs (IQPs). Several methods with warm-start strategies can be applied on IQPs, such as the interior-point methods and active-set methods. Within the FOTD scheme, the exact augmented Lagrangian function (3.4) should also be adapted to the one that can handle inequality constraints. See [47] for a particular choice. We should mention that the alternative designs with the same flavor of FOTD are also available for dealing with inequality constraints. For example, we can exploit an active-set SQP scheme, where in each iteration we only consider the inequality constraints that are in the identified active set and regard them as equalities. Then, the full problem (3.8) and the subproblem (3.9) are still equality-constrained QPs (EQPs), which are easier to solve than IQPs. Since the design and analysis of inequality constraints are quite involved, we leave the above extensions of FOTD to the future, and mainly focus on connecting FOTD with the Schwarz scheme on equality-constrained problems in this paper.

REMARK 3.6 **(relationships to other schemes)**. Besides the Schwarz scheme, FOTD is related to several other methods for optimal control problems. We introduce the connections in this remark. We emphasize that the two critical components of FOTD, overlapping temporal decomposition and augmented Lagrangian merit function, have not been investigated in the following methods. Also, the convergence of FOTD highly relies on the sensitivity analysis of NLDPs in [34, 38], which differs from the following methods.

(a) Direct multiple shooting methods. The direct multiple shooting methods decompose the full horizon into multiple *exclusive* short intervals, construct the subproblems associated with short

intervals, and compose the solution trajectories of subproblems under certain matching conditions. See [4, 5] for more details. The multiple shooting methods are often employed for real-time nonlinear model predictive control [17, 29, 50], where the subproblems are solved sequentially and an approximate but fast control feedback is desired for each subproblem. Many efficient solvers that exploit the problem sparsity structures can be used within the methods, such as qpDUNES [24] and HPMPC [25]. Compared to the aforementioned literature, FOTD solves (1.1) in an offline fashion (i.e., the short intervals do not vary with the sampling time) and applies an *overlapping* decomposition on the SQP problem instead of on (1.1). More importantly, as analyzed in section 5, FOTD does not use any matching conditions to compose the subproblem solutions, but requires a large overlap size b. Such a design is inspired by the sensitivity analysis of NLDPs. The exclusive stages are far from the boundaries where the system perturbations occur, and the sensitivity results state that the effects of boundary perturbations decay exponentially fast away from the boundaries. Thus, the solutions at the exclusive middle stages are still accurate enough even without matching conditions.

(b) Alternating direction method of multipliers (ADMM). ADMM is another popular method for optimal control problems [7, 42], where one introduces a set of consensus constraints to split the full problem into multiple subproblems, and solves the subproblems in each iteration in parallel. We note that our terminal cost (2.3) is conceptually similar to the quadratic proximal term in [42, (3)]. However, FOTD and ADMM have a few key differences. First, ADMM is designed based on the augmented Lagrangian method, while FOTD is designed based on SQP. Thus, their primal-dual updating schemes are quite different. Second, even if we use an augmented Lagrangian merit function in FOTD, the function is different from the one in ADMM [7] in that it has a quadratic penalty on the optimality error (i.e., the last term in (3.4)). Also, we use the augmented Lagrangian for the stepsize selection, not for the direction computation, while ADMM combines the two steps together. Third, similar to the multiple shooting methods, ADMM decomposes the full problem by introducing extra consensus constraints without overlaps. Thus, ADMM is sensitive to the parameter μ, while the overlaps in FOTD largely suppress the boundary perturbations brought by different choices of μ (and imprecise boundary variables $\boldsymbol{d}_i$). See [38, Figure 6] for empirical evidence.

(c) Iterative linear-quadratic regulator (ILQR). Another method for optimal control problems is ILQR [32]. In each step, the search direction of control variables is solved from a QP that is defined by quadratic approximation of objective (1.1a) with linear approximations of constraints (1.1b). A parallel implementation of ILQR was designed in [30]. In addition to the difference that FOTD employs an overlapping decomposition for parallelism, the QP in (3.8) also differs from the one in ILQR in that its objective is a quadratic approximation of the Lagrangian. Further, FOTD updates the state, control, and dual variables with the same stepsize selected by the line search on the augmented Lagrangian (3.4), while ILQR updates the control variables with the line search on the objective (1.1a), and the state variables are computed by applying (1.1b) in a forward pass.

REMARK 3.7 **(QP solvers for the subproblems)**. The subproblems (3.9) of FOTD are QPs. Many efficient QP or linear system solvers with wart-start strategies can be adopted for FOTD. For example, the conjugate gradient (CG) and minimal residual (MINRES) are popular methods for solving the Newton systems, and the popular solvers qpDUNES [24], HPMPC [25], and FORCES [19] that exploit the sparsity structures of QPs can also be applied on (3.9). We note that the above QP solvers generally require a positive definite Hessian matrix which (3.9) does not have. Fortunately, [57] proposed a convexification procedure to resolve this issue.

4. Error of the approximate search direction We study the difference between $(\tilde{\Delta}\boldsymbol{z}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau)$ and $(\Delta\boldsymbol{z}^\tau, \Delta\boldsymbol{\lambda}^\tau)$, where the former is the OTD approximation (Step 2) and the latter is the exact Newton direction of (3.2). We state the assumptions that are required for the analysis.

ASSUMPTION 4.1 **(lower bound on the reduced Hessian)**. *For any iteration $\tau \geq 0$, we let Z^τ be a matrix whose columns are orthonormal vectors and span the null space $\{\boldsymbol{w} : G^\tau \boldsymbol{w} = \boldsymbol{0}\}$. We assume that there exists a uniform constant $\gamma_{RH} > 0$ that is independent of τ, such that*

$$(Z^\tau)^T \hat{H}^\tau Z^\tau \succeq \gamma_{RH} \cdot I. \tag{4.1}$$

The matrix on the left hand side is called the reduced Hessian.

ASSUMPTION 4.2 **(controllability)**. *For any stage $k \in [N]$ and an integer $t \in [N-k]$, the controllability matrix is defined as*

$$\Xi_{k,t}(\boldsymbol{z}_{k:k+t-1}) = \left(B_{k+t-1} \;\; A_{k+t-1}B_{k+t-2} \;\cdots\; (\textstyle\prod_{l=1}^{t-1} A_{k+l})B_k\right) \in \mathbb{R}^{n_x \times t n_u}.$$

For any iteration $\tau \geq 0$, we assume that there exist a uniform constant $\gamma_C > 0$ and an integer $t > 0$ that are independent of τ, such that for any $k \in [N-t]$, there exists $t_k^\tau \in [1, t]$ so that $\Xi^\tau_{k,t_k^\tau}(\Xi^\tau_{k,t_k^\tau})^T \succeq \gamma_C I$, where $\Xi^\tau_{k,t_k^\tau} = \Xi_{k,t_k^\tau}(\boldsymbol{z}^\tau_{k:k+t_k^\tau-1})$.

ASSUMPTION 4.3 **(upper boundedness)**. *For any iteration $\tau \geq 0$, there exists a uniform constant Υ_{upper} that is independent of τ, such that for any k, $\max\{\|\hat{H}_k^\tau\|, \|A_k^\tau\|, \|B_k^\tau\|\} \leq \Upsilon_{upper}$.*

Since $G^\tau = \nabla_{\boldsymbol{z}}^T f(\boldsymbol{z}^\tau)$ has full row rank, Assumption 4.1 ensures that Problem (3.8) has a unique global solution and the KKT matrix in (3.2) is invertible [41, Lemma 16.1]. This assumption is standard in the SQP literature [3] and weaker than the linear-quadratic convex DP setup studied in [65]. Assumption 4.2 is specifically used for DP problems [28, 34, 35, 38, 65, 66]. It ensures that the linearized dynamical system $\boldsymbol{p}_{k+1} = A_k \boldsymbol{p}_k + B_k \boldsymbol{q}_k$ is controllable in at most t steps (in each iteration). That is, given any initial state $\boldsymbol{p}_k$ and target state $\boldsymbol{p}_{k+t_k}$, we can always evolve from $\boldsymbol{p}_k$ to $\boldsymbol{p}_{k+t_k}$ by specifying a suitable control sequence $\{\boldsymbol{q}_j\}_{j=k}^{k+t_k-1}$. Assumption 4.3 is standard in both SQP and DP literature [3, 34, 65]. In section 5, we will impose a compactness condition on the SQP iterates, which naturally implies the upper boundedness of the Hessian and Jacobian matrices. We show in Lemma 4.1 that Assumptions 4.2 and 4.3 imply a uniform lower bound on $G^\tau (G^\tau)^T$.

LEMMA 4.1. *Suppose $\max\{\|A_k^\tau\|, \|B_k^\tau\|\} \leq \Upsilon_{upper}$, then Assumption 4.2 implies that $G^\tau (G^\tau)^T \succeq \gamma_G I$ where*

$$\gamma_G = \gamma_G(\gamma_C, t, \Upsilon_{upper}) := \left(\frac{\gamma_C}{\gamma_C + \frac{\Upsilon_{upper}^{t+1}}{\Upsilon_{upper}-1}}\right)^2 \cdot \frac{\min\{1, \gamma_C\}}{(1+\Upsilon_{upper})^{2t}}. \tag{4.2}$$

Proof. See Appendix B.1. □

The next result shows that $\mathcal{LP}_\mu^i(\boldsymbol{d}_i)$ has a unique solution if μ is large enough.

LEMMA 4.2. *Suppose Assumptions 4.1-4.3 hold for Problem (3.8). Let*

$$\bar{\mu} = \bar{\mu}(\gamma_C, t, \Upsilon_{upper}) := \frac{32\Upsilon_{upper}^{4t+1}}{\gamma_C}. \tag{4.3}$$

If $\mu \geq \bar{\mu}$, then the reduced Hessian of Problem $\mathcal{LP}_\mu^i(\boldsymbol{d}_i)$ in (3.9), defined similarly to (4.1), is lower bounded by $\gamma_{RH} I$ for any iteration $\tau \geq 0$ and any boundary variables $\boldsymbol{d}_i$. This implies that $\mathcal{LP}_\mu^i(\boldsymbol{d}_i)$ has a unique global solution.

Proof. See Appendix B.2. □

Lemma 4.2 shows that $\{\mathcal{LP}_\mu^i(\boldsymbol{d}_i)\}_{i=0}^{M-1}$ are solvable for any iteration $\tau \geq 0$ if $\mu \geq \bar{\mu}$, where $\bar{\mu}$ is independent of the iteration index τ and the subproblem index i. An immediate consequence is that Assumptions 4.1-4.3 hold for the subproblems $\mathcal{LP}_\mu^i$ as well.

COROLLARY 4.1. *Suppose Assumptions 4.1-4.3 hold for Problem (3.8) and $\mu \geq \bar{\mu}$ with $\bar{\mu}$ given by (4.3). Then, the three conditions: lower bound on the reduced Hessian, controllability, and upper boundedness, hold for the subproblems $\{\mathcal{LP}^i_\mu(\boldsymbol{d}_i)\}_{i\in[M-1]}$ with any $\boldsymbol{d}_i$. Furthermore, the condition constants are independent of i and τ. Specifically, the subproblem $\mathcal{LP}^i_\mu(\boldsymbol{d}_i)$ for $i \in [M-1]$ satisfies*
(i) Assumption 4.1: the reduced Hessian of $\mathcal{LP}^i_\mu(\boldsymbol{d}_i)$ is lower bounded by $\gamma_{RH} I$.
(ii) Assumption 4.2: the controllability condition of $\mathcal{LP}^i_\mu(\boldsymbol{d}_i)$ is satisfied with the same constants (γ_C, t).
(iii) Assumption 4.3: the boundedness condition of $\mathcal{LP}^i_\mu(\boldsymbol{d}_i)$ is satisfied with constant $\Upsilon_{upper} + \mu$.

Proof. See Appendix B.3. □

We are now able to control the error $(\tilde{\Delta}\boldsymbol{z}^\tau - \Delta\boldsymbol{z}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau - \Delta\boldsymbol{\lambda}^\tau)$. To ease the notation, we suppress the iteration index τ. The study of the error $(\tilde{\Delta}\boldsymbol{z} - \Delta\boldsymbol{z}, \tilde{\Delta}\boldsymbol{\lambda} - \Delta\boldsymbol{\lambda})$ relies on the primal-dual sensitivity analysis of NLDPs in [34, 38], which requires Assumptions 4.1-4.3. We apply the sensitivity results on the subproblems; thus Corollary 4.1 is critical. It shows that Assumptions 4.1-4.3 hold for the subproblems as long as they hold for the full problem. In principle, the sensitivity results suggest that, if we perturb the objective and constraints on one stage, then the perturbation effects on the optimal solution decay exponentially fast away from that perturbed stage. In the OTD setup, the perturbations occur at the two boundaries of the extended intervals. Thus, the composition that uses only the exclusive part $[n_i, n_{i+1})$ of the solution preserves all accurate variables.

Let us first bound $(\tilde{\boldsymbol{w}}^\star_i(\boldsymbol{d}_i), \tilde{\boldsymbol{\zeta}}^\star_i(\boldsymbol{d}_i)) - (\tilde{\boldsymbol{w}}^\star_i(\boldsymbol{d}'_i), \tilde{\boldsymbol{\zeta}}^\star_i(\boldsymbol{d}'_i))$ for any $\boldsymbol{d}_i, \boldsymbol{d}'_i$. Recall that $(\tilde{\boldsymbol{w}}^\star_i(\boldsymbol{d}_i), \tilde{\boldsymbol{\zeta}}^\star_i(\boldsymbol{d}_i))$ denotes the unique global solution of $\mathcal{LP}^i_\mu(\boldsymbol{d}_i)$. In the following presentation, we use C and C' to denote universal constants that are independent of the iteration index τ and the subproblem index i.

THEOREM 4.1. *Let Assumptions 4.1-4.3 hold for Problem (3.8) and $\mu \geq \bar{\mu}$ with $\bar{\mu}$ given by (4.3). There exist constants $C' > 0$ and $\rho \in (0,1)$ independent of i and τ, such that for all $k \in [m_1, m_2]$,*

$$\max\{\|\tilde{\boldsymbol{w}}^\star_{i,k}(\boldsymbol{d}_i) - \tilde{\boldsymbol{w}}^\star_{i,k}(\boldsymbol{d}'_i)\|,\ \|\tilde{\boldsymbol{\zeta}}^\star_{i,k}(\boldsymbol{d}_i) - \tilde{\boldsymbol{\zeta}}^\star_{i,k}(\boldsymbol{d}'_i)\|\} \leq C'(\rho^{k-m_1}\|\boldsymbol{d}_{i,1} - \boldsymbol{d}'_{i,1}\| + \rho^{m_2-k}\|\boldsymbol{d}_{i,2:4} - \boldsymbol{d}'_{i,2:4}\|)$$

for any two boundary variables $\boldsymbol{d}_i$ and $\boldsymbol{d}'_i$.

Proof. See Appendix B.4. □

The next theorem characterizes the approximation error of the Newton direction.

THEOREM 4.2 **(error of approximate direction)**. *Let Assumptions 4.1-4.3 hold for Problem (3.8) and $\mu \geq \bar{\mu}$ with $\bar{\mu}$ given by (4.3). There exist constants $C > 0$ and $\rho \in (0,1)$ independent of τ, such that*

$$\|(\tilde{\Delta}\boldsymbol{z}^\tau - \Delta\boldsymbol{z}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau - \Delta\boldsymbol{\lambda})\| \leq C\rho^b\|(\Delta\boldsymbol{z}^\tau, \Delta\boldsymbol{\lambda}^\tau)\|. \tag{4.4}$$

Proof. See Appendix B.5. □

Theorem 4.2 suggests that the error of the approximate direction decays exponentially fast in terms of the overlap size b. We can naturally expect that $(\tilde{\Delta}\boldsymbol{\lambda}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau)$ contains enough information to decrease the merit function in each iteration, provided b is large. We study how the inexactness of the direction affects SQP in the next section.

5. Global convergence of FOTD To enable general distributed methods for solving (3.2), we suppose that a decomposition method outputs a direction $(\tilde{\Delta}\boldsymbol{z}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau)$ satisfying

$$\|(\tilde{\Delta}\boldsymbol{z}^\tau - \Delta\boldsymbol{z}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau - \Delta\boldsymbol{\lambda}^\tau)\| \leq \delta \cdot \|(\Delta\boldsymbol{z}^\tau, \Delta\boldsymbol{\lambda}^\tau)\|, \quad \text{for } \delta \in (0,1). \tag{5.1}$$

Theorem 4.2 shows that FOTD specializes (5.1) with $\delta = C\rho^b$. In this section, we study the convergence of SQP with the merit function (3.4) and the direction $(\tilde{\Delta}\boldsymbol{z}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau)$. We require a compactness assumption to strengthen Assumption 4.3 to hold in a compact set. We recall that the SQP iterates are generated by (3.3) and (3.5).

ASSUMPTION 5.1 **(compactness)**. *There exists a compact set $\mathcal{Z} \times \Lambda$, where $\mathcal{Z} = \mathcal{Z}_0 \times \cdots \times \mathcal{Z}_N$ and $\Lambda = \Lambda_0 \times \cdots \times \Lambda_N$, such that $(\boldsymbol{z}^\tau + \alpha\tilde{\Delta}\boldsymbol{z}^\tau, \boldsymbol{\lambda}^\tau + \alpha\tilde{\Delta}\boldsymbol{\lambda}^\tau) \in \mathcal{Z} \times \Lambda$, $\forall \tau \geq 0, \alpha \in [0,1]$; and we assume that $\{g_k, f_k\}$ are thrice continuously differentiable and*

$$\max\{\sup_{\mathcal{Z}_k \times \Lambda_{k+1}} \|H_k(\boldsymbol{z}_k, \boldsymbol{\lambda}_{k+1})\|, \sup_{\mathcal{Z}_k} \|A_k(\boldsymbol{z}_k)\|, \sup_{\mathcal{Z}_k} \|B_k(\boldsymbol{z}_k)\|\} \leq \Upsilon_{upper}, \tag{5.2}$$

for a constant $\Upsilon_{upper} > 0$ independent of k.

We use the same constant Υ_{upper} as in Assumption 4.3 for notational simplicity. The compactness assumption is standard in the SQP literature [3, Proposition 4.15]. One equivalent assumption is to assume that the sublevel set $\{(\boldsymbol{z}, \boldsymbol{\lambda}) : \mathcal{L}_\eta(\boldsymbol{z}, \boldsymbol{\lambda}) \leq \mathcal{L}_\eta^0\}$ is contained in a compact set. Furthermore, assuming the existence of the third derivatives on $\{g_k, f_k\}$ is common for the augmented Lagrangian merit function (3.4), since $\nabla^2\mathcal{L}_\eta$ requires $\nabla^3 g_k$ and $\nabla^3 f_k$ [3, 36, 37, 69]. Note that the third derivatives are only required in the analysis, and not computed or used in the algorithm.

The following lemma is an immediate consequence of Assumption 5.1.

LEMMA 5.1. *Under Assumption 5.1, there exists a constant $\Upsilon_{HG} > 0$ such that*

$$\max\{\sup_{\mathcal{Z} \times \Lambda} \|H(\boldsymbol{z}, \boldsymbol{\lambda})\|, \sup_{\mathcal{Z}} \|G(\boldsymbol{z})\|\} \leq \Upsilon_{HG}. \tag{5.3}$$

Further, if Assumptions 4.1-4.3 hold as well, there exists a constant $\Upsilon_{KKT} > 0$ that is independent of τ, such that

$$\left\| \begin{pmatrix} \hat{H}^\tau & (G^\tau)^T \\ G^\tau & \mathbf{0} \end{pmatrix}^{-1} \right\| \leq \Upsilon_{KKT}, \quad \forall \tau \geq 0.$$

Proof. See Appendix C.1 □

The bound (5.3) suggests that $\|G^\tau\| \leq \Upsilon_{HG}, \forall \tau \geq 0$ (since $\boldsymbol{z}^\tau \in \mathcal{Z}$). Since (5.2) implies $\max\{\|A_k^\tau\|, \|B_k^\tau\|\} \leq \Upsilon_{upper}$ for any $k \in [N-1]$, under Assumption 5.1, we only need $\|\hat{H}^\tau\| \leq \Upsilon_{upper}$ from Assumption 4.3. To simplify the presentation, we define $\Upsilon = \max\{\Upsilon_{upper}, \Upsilon_{KKT}, \Upsilon_{HG}\}$.

We now show that $(\tilde{\Delta}\boldsymbol{z}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau)$ is a descent direction of $\mathcal{L}_\eta^\tau$ provided η_1 is sufficiently large and η_2, δ are sufficiently small.

THEOREM 5.1. *Suppose Assumptions 4.1, 4.2, 4.3, 5.1 hold for the SQP iterates with a search direction $(\tilde{\Delta}\boldsymbol{z}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau)$ satisfying (5.1). If*

$$\eta_1 \geq \frac{17}{\eta_2 \gamma_G}, \quad \eta_2 \leq \frac{\gamma_{RH}}{12\Upsilon^2}, \quad \delta \leq \frac{\eta_2 \gamma_G}{9\eta_1 \Upsilon^2}, \tag{5.4}$$

where γ_G is defined in (4.2), then

$$\begin{pmatrix} \nabla_{\boldsymbol{z}} \mathcal{L}_\eta^\tau \\ \nabla_{\boldsymbol{\lambda}} \mathcal{L}_\eta^\tau \end{pmatrix}^T \begin{pmatrix} \tilde{\Delta}\boldsymbol{z}^\tau \\ \tilde{\Delta}\boldsymbol{\lambda}^\tau \end{pmatrix} \leq -\frac{\eta_2}{2} \left\| \begin{pmatrix} \nabla_{\boldsymbol{z}} \mathcal{L}^\tau \\ \nabla_{\boldsymbol{\lambda}} \mathcal{L}^\tau \end{pmatrix} \right\|^2. \tag{5.5}$$

Proof. See Appendix C.2. □

By the results of Theorem 5.1 and the mean value theorem, we immediately know that, under the presented conditions, the stepsize α_τ to satisfy the Armijo condition (3.5) can be found by the backtracking line search, and the SQP iterates with the direction $(\tilde{\Delta}\boldsymbol{z}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau)$ make successful progress towards a stationary point.

Theorem 5.1 also shows the importance of using the exact augmented Lagrangian merit function (3.4). In particular, as proved in (C.8), we rely on a critical property: for suitably chosen $\eta = (\eta_1, \eta_2)$, there exists a small constant $\kappa(\eta_2) > 0$ (depending on η_2) such that

$$\begin{pmatrix} \nabla_{\boldsymbol{z}} \mathcal{L}_\eta^\tau \\ \nabla_{\boldsymbol{\lambda}} \mathcal{L}_\eta^\tau \end{pmatrix}^T \begin{pmatrix} \Delta \boldsymbol{z}^\tau \\ \Delta \boldsymbol{\lambda}^\tau \end{pmatrix} \leq -\kappa(\eta_2) \Bigg(\underbrace{\left\| \begin{pmatrix} \nabla_{\boldsymbol{z}} \mathcal{L}^\tau \\ \nabla_{\boldsymbol{\lambda}} \mathcal{L}^\tau \end{pmatrix} \right\|^2}_{\text{ensure convergence}} + \underbrace{\left\| \begin{pmatrix} \Delta \boldsymbol{z}^\tau \\ \Delta \boldsymbol{\lambda}^\tau \end{pmatrix} \right\|^2}_{\text{allow approximation}} \Bigg). \tag{5.6}$$

The first term is needed for global convergence when we accumulate the descent over iterations. The second term allows us to replace $(\Delta \boldsymbol{z}^\tau, \Delta \boldsymbol{\lambda}^\tau)$ by $(\tilde{\Delta}\boldsymbol{z}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau)$ because, as proved in (C.10),

$$\left\| \begin{pmatrix} \nabla_{\boldsymbol{z}} \mathcal{L}_\eta^\tau \\ \nabla_{\boldsymbol{\lambda}} \mathcal{L}_\eta^\tau \end{pmatrix} \right\| \left\| \begin{pmatrix} \tilde{\Delta}\boldsymbol{z}^\tau - \Delta \boldsymbol{z}^\tau \\ \tilde{\Delta}\boldsymbol{\lambda}^\tau - \Delta \boldsymbol{\lambda}^\tau \end{pmatrix} \right\| \lesssim \delta \left\| \begin{pmatrix} \Delta \boldsymbol{z}^\tau \\ \Delta \boldsymbol{\lambda}^\tau \end{pmatrix} \right\|^2,$$

where $\lesssim$ means that the inequality holds up to a constant multiplier. Therefore, for small enough δ, the margin $\|(\Delta \boldsymbol{z}^\tau, \Delta \boldsymbol{\lambda}^\tau)\|$ in (5.6) allows for an approximation error. Our analysis rules out the exact penalty merit functions that depend only on the primal variables $\boldsymbol{z}$. Such a function $\mathcal{M}(\boldsymbol{z})$ satisfies $\nabla_{\boldsymbol{z}}^T \mathcal{M}^\tau \Delta \boldsymbol{z}^\tau \leq -\kappa \|\Delta \boldsymbol{z}^\tau\|^2$ and $\|\nabla_{\boldsymbol{z}} \mathcal{M}^\tau\| \lesssim \|\Delta \boldsymbol{z}^\tau\|$ (e.g., [3, 22, 26, 44, 49]). If we approximate $\Delta \boldsymbol{z}^\tau$ by $\tilde{\Delta}\boldsymbol{z}^\tau$, Theorems 4.1, 4.2 suggest that the approximation error $\|\tilde{\Delta}\boldsymbol{z}^\tau - \Delta \boldsymbol{z}^\tau\|$ also depends on $\|\Delta \boldsymbol{\lambda}^\tau\|$. Thus, the margin $\|\Delta \boldsymbol{z}^\tau\|^2$ may not be enough to endure the approximation error. This reveals the benefits of using the primal-dual exact merit functions based on (3.4).

We summarize the global convergence results in the next theorem.

THEOREM 5.2 **(global convergence)**. *Suppose Assumptions 4.1, 4.2, 4.3, 5.1 hold for the SQP iterates with $\{(\tilde{\Delta}\boldsymbol{z}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau)\}_\tau$ satisfying (5.1) and parameters (η, δ) satisfying (5.4), then $\|\nabla \mathcal{L}^\tau\| \to 0$ as $\tau \to \infty$. Furthermore, for the FOTD iterates, if $\mu \geq \bar{\mu}$ with $\bar{\mu}$ given by (4.3) as well as b satisfies*

$$b \geq \frac{\log\left(9C\eta_1 \Upsilon^2 / (\eta_2 \gamma_G)\right)}{\log(1/\rho)} \tag{5.7}$$

with $C > 0$ and $\rho \in (0, 1)$ from Theorem 4.2, then the FOTD iterates $\{(\boldsymbol{z}^\tau, \boldsymbol{\lambda}^\tau)\}_\tau$ generated by Algorithm 2 satisfy $\|\nabla \mathcal{L}^\tau\| \to 0$ as $\tau \to \infty$.

Proof. See Appendix C.3. □

We have shown the global convergence of FOTD. Theorem 5.2 complements the local convergence result of the Schwarz scheme [38] and answers Q1 raised in section 1. The next section studies the local convergence of FOTD. We show that FOTD and the Schwarz have the same local behavior.

To end this section, we discuss an adaptivity extension of our scheme.

REMARK 5.1 **(adaptivity on penalty parameters)**. Given the conditions (5.4) on $(\eta, \delta = C\rho^b)$, it is possible to design a scheme that adaptively selects the suitable penalty parameters η and overlap size b. Since the upper/lower bound constants in (5.4) do not depend on τ, we can design a while loop to achieve this goal. In particular, we let $\nu > 1$ be fixed. While (5.5) does not hold, we let

$$\eta_2 \leftarrow \eta_2 / \nu, \quad \eta_1 \leftarrow \eta_1 \nu^2, \quad \delta \leftarrow \delta / \nu^4.$$

Then, we know $\eta_1 \eta_2$ increases by a factor of ν, and η_2 and $\delta \eta_1 / \eta_2$ decrease by a factor of $1/\nu$. Thus, for sufficiently large τ, all parameters are stabilized. We also note that $\delta \leftarrow \delta / \nu^4$ is equivalent to letting $b \leftarrow b + 4 \log \nu / \log(1/\rho)$, where only ρ has to be tuned manually.

6. Local convergence of FOTD From now on, we suppose the FOTD iterates satisfy $(\boldsymbol{z}^\tau, \boldsymbol{\lambda}^\tau) \to (\boldsymbol{z}^\star, \boldsymbol{\lambda}^\star)$ as $\tau \to \infty$. For two positive sequences $\{a_\tau\}$ and $\{b_\tau\}$, $a_\tau = O(b_\tau)$ if a_τ/b_τ is uniformly bounded over τ; $a_\tau = o(b_\tau)$ if $a_\tau/b_\tau \to 0$ as $\tau \to \infty$. Our local analysis is divided into three steps:

(a) we show that $\alpha_\tau = 1$ is selected for the Armijo condition (3.5) when τ is large.
(b) we show a relationship between FOTD and the Schwarz scheme: Line 6 in Algorithm 2 is equivalent to performing a single Newton step for subproblems in Line 5 in Algorithm 1.
(c) we prove that FOTD converges linearly, with a linear rate that decays exponentially in b.

We present additional assumptions for local analysis.

ASSUMPTION 6.1 **(Hessian approximation vanishes)**. *We assume that* $\|H^\tau - \hat{H}^\tau\| = o(1)$, *where* $H^\tau = \nabla^2_{\boldsymbol{z}} \mathcal{L}^\tau$ *is the Lagrangian Hessian and* $\hat{H}^\tau$ *is its approximation.*

A vanishing Hessian modification is typical for the local analysis of SQP to have a superlinear (or quadratic) convergence [6]. As discussed in Remark 3.4, we can check (4.1) for Hessian H^τ to decide if a modification is needed, since (4.1) holds locally if SOSC is satisfied at $(\boldsymbol{z}^\star, \boldsymbol{\lambda}^\star)$. Equivalently, we check the positive definiteness of $H^\tau + c(G^\tau)^T G^\tau$ for a constant c, which is a block-tridiagonal matrix. A parallel Cholesky decomposition is applicable in this regard.

ASSUMPTION 6.2 **(local Lipschitz continuity)**. *We assume there exists a constant* Υ_L *independent of* k *such that, for any two points* $(\boldsymbol{z}, \boldsymbol{\lambda})$ *and* $(\boldsymbol{z}', \boldsymbol{\lambda}')$ *sufficiently close to* $(\boldsymbol{z}^\star, \boldsymbol{\lambda}^\star)$,

$$\begin{aligned}\max\{\|A_k(\boldsymbol{z}_k) - A_k(\boldsymbol{z}'_k)\|,\, \|B_k(\boldsymbol{z}_k) - B_k(\boldsymbol{z}'_k)\|\} &\le \Upsilon_L \|\boldsymbol{z}_k - \boldsymbol{z}'_k\|, \\ \|H_k(\boldsymbol{z}_k, \boldsymbol{\lambda}_{k+1}) - H_k(\boldsymbol{z}'_k, \boldsymbol{\lambda}'_{k+1})\| &\le \Upsilon_L \left\|(\boldsymbol{z}_k - \boldsymbol{z}'_k; \boldsymbol{\lambda}_{k+1} - \boldsymbol{\lambda}'_{k+1})\right\|.\end{aligned}$$

Assumption 6.2 strengthens the boundedness condition in Assumption 5.1 to the (local) Lipschitz continuity. We start the local analysis with Step (a).

Step (a): A unit stepsize is accepted. We have the following theorem.

THEOREM 6.1. *Suppose Assumptions 4.1, 4.2, 4.3, 5.1, 6.1, 6.2 hold for the SQP iterates with search directions* $\{(\tilde{\Delta}\boldsymbol{z}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau)\}_\tau$ *satisfying* (5.1) *and parameters* (η, δ) *satisfying*

$$\eta_1 \ge \frac{17}{\eta_2 \gamma_G}, \quad \eta_2 \le \frac{\gamma_{RH}}{12\Upsilon^2}, \quad \delta \le \frac{1/2 - \beta}{3/2 - \beta} \cdot \frac{\eta_2 \gamma_G}{9\eta_1 \Upsilon^2}, \tag{6.1}$$

then $\alpha_\tau = 1$ *for all sufficiently large* τ.

Proof. See Appendix D.1. □

The condition (6.1) on δ is stronger than (5.4) up to a multiplier depending on β. For FOTD, we suppose $\mu \ge \bar{\mu}$ with $\bar{\mu}$ given by (4.3), and let $\delta = C\rho^b$ in (6.1) to get a condition on b as in (6.3).

Step (b): A relation between FOTD and the Schwarz scheme. We establish a relationship between FOTD and the Schwarz scheme. We will show that, by specifying $\boldsymbol{d}^\tau = (\mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0})$ in Problem (3.9), FOTD is equivalent to performing *one Newton step* for subproblems (2.2) with a *warm-start initialization* in the Schwarz scheme. The result is summarized in the next theorem.

THEOREM 6.2. *Given the current iterate* $(\boldsymbol{z}^\tau, \boldsymbol{\lambda}^\tau)$, *we consider two procedures:*
(a) Schwarz with a warm-start initialization: for $i \in [M-1]$, *we specify boundary variables* $\boldsymbol{d}_i^\tau = (\boldsymbol{x}_{m_1}^\tau; \boldsymbol{x}_{m_2}^\tau; \boldsymbol{u}_{m_2}^\tau; \boldsymbol{\lambda}_{m_2+1}^\tau)$; *perform* ***one full Newton step*** *for* $\mathcal{P}_\mu^i(\boldsymbol{d}_i^\tau)$ (2.2) *at* $\mathcal{D}_i(\boldsymbol{z}^\tau, \boldsymbol{\lambda}^\tau)$ *and get* $(\tilde{\boldsymbol{z}}_i^{\tau+1}, \tilde{\boldsymbol{\lambda}}_i^{\tau+1})$; *then let* $(\boldsymbol{z}^{\tau+1}, \boldsymbol{\lambda}^{\tau+1}) = \mathcal{C}(\{(\tilde{\boldsymbol{z}}_i^{\tau+1}, \tilde{\boldsymbol{\lambda}}_i^{\tau+1})\}_i)$.
(b) FOTD scheme without Hessian approximation: for $i \in [M-1]$, *we specify boundary variables* $\boldsymbol{d}_i^\tau = (\mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0})$; *solve* $\mathcal{LP}_\mu^i(\boldsymbol{d}_i^\tau)$ *in* (3.9) *with* $\hat{H}_k^\tau = H_k^\tau$, $\forall k \in [m_1, m_2]$; *obtain* $(\tilde{\boldsymbol{w}}_i^\star(\boldsymbol{d}_i^\tau), \tilde{\boldsymbol{\zeta}}_i^\star(\boldsymbol{d}_i^\tau))$; *then let* $(\tilde{\Delta}\boldsymbol{z}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau) = \mathcal{C}(\{(\tilde{\boldsymbol{w}}_i^\star(\boldsymbol{d}_i^\tau), \tilde{\boldsymbol{\zeta}}_i^\star(\boldsymbol{d}_i^\tau))\}_i)$ *and update as* $(\boldsymbol{z}^{\tau+1}, \boldsymbol{\lambda}^{\tau+1}) = (\boldsymbol{z}^\tau, \boldsymbol{\lambda}^\tau) + (\tilde{\Delta}\boldsymbol{z}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau)$.
Then, starting from $(\boldsymbol{z}^\tau, \boldsymbol{\lambda}^\tau)$, *both procedures generate the same next iterate* $(\boldsymbol{z}^{\tau+1}, \boldsymbol{\lambda}^{\tau+1})$.

Proof. See Appendix D.2. □

Theorem 6.2 reveals a strong relation between the Schwarz scheme and FOTD. Locally, FOTD can be seen as an improvement of the warm-start Schwarz scheme, where a single Newton step is performed for the subproblems, instead of solving the subproblems to optimality as the original Schwarz did. The warm initialization is recommended by [38] for practical purpose, which avoids the case where the solutions of the same subproblem in different iterations are very distinct.

Step (c): local linear convergence of FOTD. We now establish the local convergence rate for FOTD. For $\tau \geq 0$ we let

$$\Psi^\tau := \max_{k\in[N]} \Psi_k^\tau := \max_{k\in[N]} \|(\boldsymbol{z}_k^\tau, \boldsymbol{\lambda}_k^\tau) - (\boldsymbol{z}_k^\star, \boldsymbol{\lambda}_k^\star)\|. \tag{6.2}$$

We require the following lemma that shows the one-step error recursion. We use C_1, C_2 to denote generic constants that are ensured to exist, but may differ from the constant C in Theorem 4.2. The constant ρ is from Theorem 4.2.

LEMMA 6.1. *Consider the FOTD iterates $\{(\boldsymbol{z}^\tau, \boldsymbol{\lambda}^\tau)\}_\tau$ under Assumptions 4.1, 4.2, 4.3, 5.1, 6.1, 6.2 and suppose $\mu \geq \bar{\mu}$ with $\bar{\mu}$ given by (4.3) and η satisfies (6.1). There exist constants $C_1 > 0$ and $\rho \in (0,1)$ independent of $\tau, \eta_1, \eta_2, \beta$, such that, if b satisfies*

$$b \geq \frac{\log\{(3/2-\beta)C_1\eta_1/((1/2-\beta)\eta_2)\}}{\log(1/\rho)}, \tag{6.3}$$

then, for all sufficiently large τ (in each case below, m_1, m_2 depend on i correspondingly, see (2.1)),

$$\Psi_k^{\tau+1} \leq o(\Psi^\tau) + C_1\left\{\rho^{k-m_1}\|\boldsymbol{x}_{m_1}^\tau - \boldsymbol{x}_{m_1}^\star\| + \rho^{m_2-k}\left\|\begin{pmatrix}\boldsymbol{z}_{m_2}^\tau - \boldsymbol{z}_{m_2}^\star \\ \boldsymbol{\lambda}_{m_2+1}^\tau - \boldsymbol{\lambda}_{m_2+1}^\star\end{pmatrix}\right\|\right\}, \quad {\scriptstyle \forall k\in[n_i,n_{i+1}) \atop \scriptstyle i\in[M-2]}, \tag{6.4a}$$

$$\Psi_k^{\tau+1} \leq o(\Psi^\tau) + C_1\rho^{k-m_1}\|\boldsymbol{x}_{m_1}^\tau - \boldsymbol{x}_{m_1}^\star\|, \quad \forall k \in [n_{M-1}, n_M]. \tag{6.4b}$$

Proof. See Appendix D.3. □

Lemma 6.1 relies on the decay structure of the KKT matrix inverse, established in [35, Lemma 2]. In particular, the authors showed that, if Assumptions 4.1-4.3 hold for subproblems (3.9) (verified in Corollary 4.1), the block matrices of KKT inverse corresponding to each stage have an exponentially decay structure (see [35, Figure 2]). The core of such a result is the primal-dual sensitivity analysis of NLDPs [34, 38], which we also made use of in the proof of Theorem 4.1.

Seeing from (6.4), the first $o(\Psi^\tau)$ term is the algorithmic convergence rate, which is superlinear and achieved by the SQP framework; the second term $C_1\{\rho^{k-m_1}\|\boldsymbol{x}_{m_1}^\tau - \boldsymbol{x}_{m_1}^\star\| + \rho^{m_2-k}\|(\boldsymbol{z}_{m_2}^\tau, \boldsymbol{\lambda}_{m_2+1}^\tau) - (\boldsymbol{z}_{m_2}^\star, \boldsymbol{\lambda}_{m_2+1}^\star)\|\}$ has a linear convergence rate, which is brought by the horizon truncation in OTD. Specifically, the perturbations come from the misspecification of boundary variables: we use $\boldsymbol{d}_i^\tau = (\mathbf{0}; \mathbf{0}; \mathbf{0}; \mathbf{0})$ while $\boldsymbol{d}_i^\star = (\Delta\boldsymbol{x}_{m_1}; \Delta\boldsymbol{x}_{m_2}; \Delta\boldsymbol{u}_{m_2}; \Delta\boldsymbol{\lambda}_{m_2+1})$.

Our result in Lemma 6.1 is different from [35, Theorem 2], where the algorithmic convergence rate is quadratic as they used the unperturbed Hessian H^τ in each iteration. The Hessian modification is necessary in our case as it ensures the global convergence. Our result is also different from [38, Theorem 7], which has no algorithmic rate as they solved the subproblems to optimality. Our result reveals that, even if we do not solve the subproblems to optimality, as long as the algorithmic rate is faster than linear (superlinear in our case), the perturbation rate will always dominate for large τ, which is linear. Therefore, a local linear convergence is still achieved.

We summarize the local convergence guarantee in the next theorem.

THEOREM 6.3 **(uniform linear convergence)**. *Under the setup of Lemma 6.1, we have for all sufficiently large τ that $\Psi^{\tau+1} \leq 3C_1\rho^b \cdot \Psi^\tau$ for constants $C_1 > 0$ and $\rho \in (0,1)$ from Lemma 6.1.*

Proof. We use the facts that $\boldsymbol{x}_0^\tau = \boldsymbol{x}_0^\star = \bar{\boldsymbol{x}}_0$ (see the proof of Theorem 4.2) and $\min\{k-m_1, m_2-k\} \geq b$ for all $k \in [n_i, n_{i+1}]$, and apply (6.4). Then, we obtain $\Psi_k^{\tau+1} \leq 3C_1\rho^b\Psi^\tau$ for all k. Thus, by (6.2), we complete the proof. □

We finally summarize our convergence analysis of FOTD in the next theorem.

THEOREM 6.4 **(convergence of FOTD in Algorithm 2).** *Consider Algorithm 2 under Assumptions 4.1-4.3, 5.1, and suppose μ satisfies (4.3). There exist constants $C_2 > 0, \rho \in (0,1)$ independent of τ and algorithmic parameters η_1, η_2, β, such that if*

$$\eta_1 \geq \frac{17}{\eta_2\gamma_G}, \quad \eta_2 \leq \frac{\gamma_{RH}}{12\Upsilon^2}, \quad b \geq \frac{\log\{(3/2-\beta)C_2\eta_1/((1/2-\beta)\eta_2)\}}{\log(1/\rho)},$$

then $\|\nabla\mathcal{L}^\tau\| \to 0$ as $\tau \to \infty$. Moreover, if Assumptions 6.1, 6.2 hold locally as well, then for all sufficiently large τ, $\alpha_\tau = 1$ and

$$\Psi^{\tau+1} \leq (C_2\rho^b)\Psi^\tau. \tag{6.5}$$

Proof. By Theorems 5.2, 6.1, 6.3, Lemma 6.1 and rescaling C_2 properly, we complete the proof. □

The result in (6.5) matches the local result (2.5) proved in [38]; thus, we answer the Q2 raised in section 1—it is not necessary to solve the subproblems to optimality for achieving the local linear convergence. As mentioned earlier, as long as the algorithmic rate is faster than linear (e.g., one Newton step for each subproblem), the local linear rate, induced by the decay of the sensitivity of perturbations, is always achieved. We should also mention that both the Schwarz scheme and FOTD have linear rates of the form $C\rho^b$. Since the analyses of both algorithms only claim the existence of some constants $C > 0$ and $\rho \in (0,1)$ (see Theorem 6.4 and [38, Theorem 8]), we can always use larger constants between the two algorithms to make their linear rates identical. In fact, comparing their constant C can be difficult since it depends on the sharpness of the derivation and various quantities of the problem (e.g., Υ, γ_G etc.). However, the constant ρ is the same, and is from [34, Theorem 5.7]. More importantly, both their linear rates decay exponentially in the overlap size b.

7. Numerical experiments We first conduct a numerical experiment on a toy NLDP in [35]:

$$\min_{\boldsymbol{x},\boldsymbol{u}} \sum_{k=0}^{N-1} \left\{2\cos(x_k - d_k)^2 + C_1(x_k-d_k)^2 - C_2(u_k-d_k)^2\right\} + C_1 x_N^2, \tag{7.1a}$$

$$\text{s.t. } x_{k+1} = x_k + u_k + d_k, \quad \forall k \in [N-1], \tag{7.1b}$$

$$x_0 = 0. \tag{7.1c}$$

Here, $n_x = n_u = 1$, and references $\{d_k\}$ are specified later. As checked in [35], if $C_1 - 2 > 4|C_2|$, then $Z^T(\boldsymbol{z})H(\boldsymbol{z},\boldsymbol{\lambda})Z(\boldsymbol{z}) \succeq (C_1 - 2 - 4|C_2|)/4 \cdot I \ \forall \boldsymbol{z}, \boldsymbol{\lambda}$. Thus, we simply let $\hat{H}^\tau = H^\tau$ in implementation.

We implement seven methods: one centralized method IPOPT [60] (which is our baseline), and six parallel methods including the proposed FOTD, the Schwarz method [38], the direct multiple shooting method [4], the iterative differential dynamic programming (IDDP) method [55], ILQR [32], and ADMM [42]. The scheme of IDDP is almost the same as ILQR, except that the control variables of QPs are computed by rolling out the policies along the original nonlinear dynamics rather than the linearized dynamics [48]. Since the dynamics (7.1b) are linear, IDDP and ILQR are identical in this case. We will study a temperature control problem of thin plates later, where we then have nonlinear dynamics. We aim to demonstrate three points in this experiment: (i) FOTD converges globally while the Schwarz may not converge *within a reasonable computational budget* for some initializations. (ii) FOTD exhibits at least linear convergence locally, and the larger overlap size b leads to the faster convergence. (iii) FOTD is a superior parallel method. It is as competitive as the centralized solver IPOPT, and robust to the penalty parameter μ (cf. (3.9a)). To illustrate the first point, we

generate random initial iterates that are shared by all methods, and see if each method converges for all initializations. To illustrate the second point, we plot $\|\nabla\mathcal{L}^\tau\|$ v.s. τ for FOTD with different b, and see how $\|\nabla\mathcal{L}^\tau\|$ behaves on the tail. To illustrate the third point, we compare the KKT residual $\|\nabla\mathcal{L}^\tau\|$ and running time for all methods, and vary μ for FOTD to test its robustness.

Simulation setting: We consider three cases in Table 1. For all parallel methods, we use the same horizon decomposition; that is, we decompose $[0, N]$ evenly with length M for each (exclusive) short interval. Different from the methods of multiple shooting, ILQR, IDDP, and ADMM, the Schwarz and FOTD have overlaps between two successive short intervals. We vary the overlap size b from $\{1, 5, 25\}$. For FOTD, we let $\eta_1 = 10$, $\eta_2 = \beta = 0.1$, and vary μ in a wide range $\{1, 25, 125\}$. The setup of μ is shared by the Schwarz and ADMM, both of which also require the penalty parameter. When doing the backtracking line search, we decrease the stepsize by a factor of 0.9 each time until the line search condition is satisfied. For each case in Table 1 and each setup of b and μ, we generate 5 initial iterates for FOTD that are shared by other methods: one is $(\boldsymbol{z}^0, \boldsymbol{\lambda}^0) = (\mathbf{0}, \mathbf{0})$ and the other four are from $x_k^0, u_k^0, \lambda_k^0 \overset{iid}{\sim} \text{Uniform}(-10^5, 10^5)$ (with $x_0^0 = 0$). For those methods that have to solve nonlinear subproblems (i.e., the Schwarz, multiple shooting, ADMM), we apply Julia/JuMP package [21] with IPOPT solver [60]. For all methods except ADMM, we stop the iteration if

$$\|\nabla\mathcal{L}^\tau\| \le 10^{-6} \qquad \text{OR} \qquad \|(\boldsymbol{z}^{\tau+1} - \boldsymbol{z}^\tau; \boldsymbol{\lambda}^{\tau+1} - \boldsymbol{\lambda}^\tau)\| \le 10^{-6}. \tag{7.2}$$

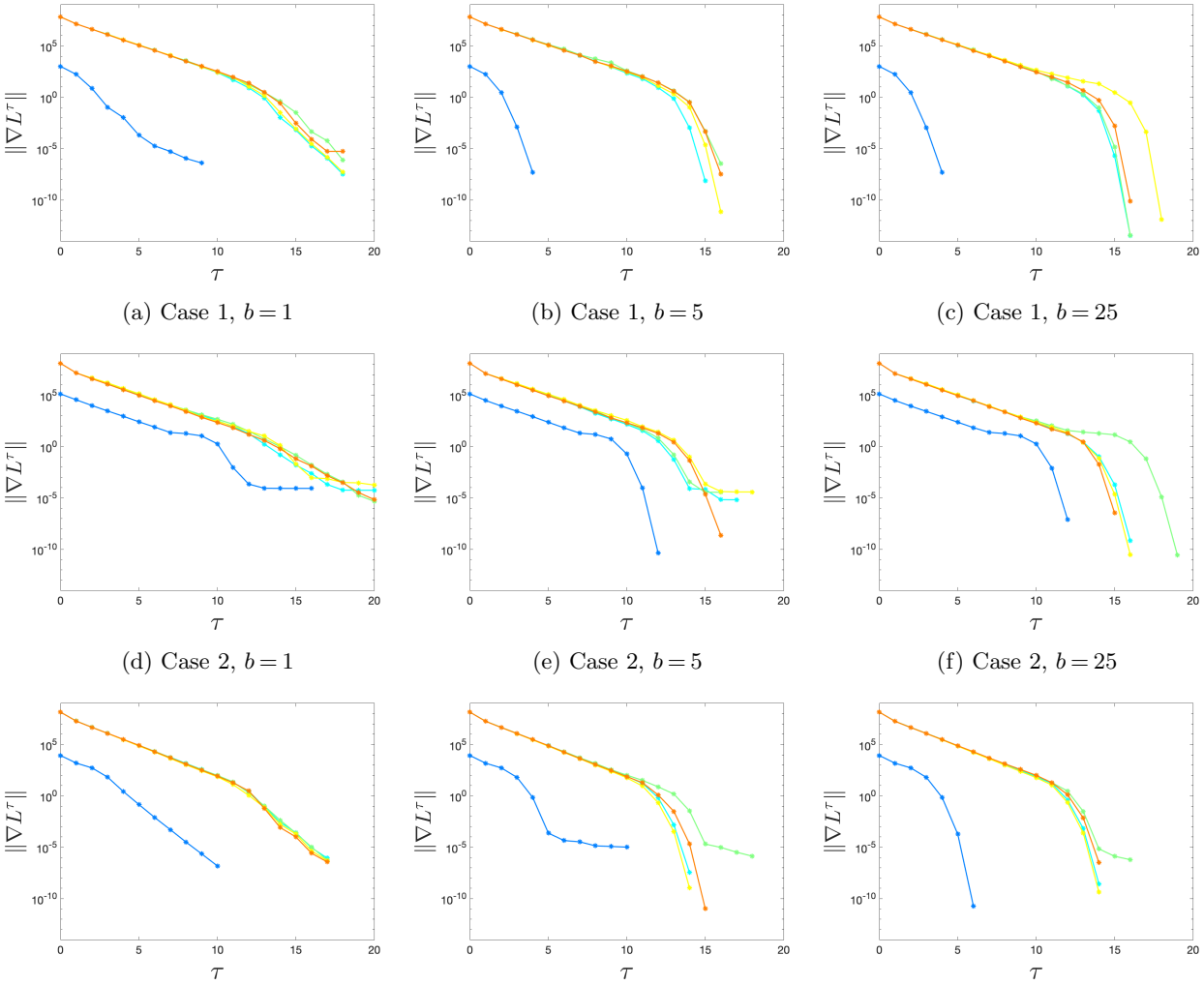
Since FOTD, ILQR, and IDDP have comparable computations in each iteration (i.e., they all solve QPs), we regard them as *converged* if they trigger the condition (7.2) within 40 iterations budget (we see from Figure 2 that FOTD actually needs much less iterations). For the multiple shooting and Schwarz that solve NLDPs, we reduce the iteration budget a bit to 30 for sake of a comparable total computation cost. For ADMM, we observe in our experiment that its KKT sequence has a long flat tail (see [38, Figure 6]). Thus, we prefer to stop the ADMM iteration early by relaxing the condition (7.2) to $\|\nabla\mathcal{L}^\tau\| \le 10^{-6}$ OR $\|(\boldsymbol{z}^{\tau+1} - \boldsymbol{z}^\tau; \boldsymbol{\lambda}^{\tau+1} - \boldsymbol{\lambda}^\tau)\| \le 10^{-3}$, and increase its iteration budget to 100. For the methods that do not compute $\boldsymbol{\lambda}$ (i.e., multiple shooting, ADMM, ILQR, IDDP), we let $\boldsymbol{\lambda}^\tau = -(G^\tau(G^\tau)^T)^{-1}G^\tau\nabla g(\boldsymbol{z}^\tau)$ when evaluating $\nabla\mathcal{L}^\tau$. Note that $\boldsymbol{\lambda}^\tau \to \boldsymbol{\lambda}^\star$ as $\boldsymbol{z}^\tau \to \boldsymbol{z}^\star$. In addition to the above settings, we try three linear system solvers for FOTD: one is sparse LU (which is a default choice and adopted by other methods), and the other two are generalized minimal residual (GMRES) and induced dimension reduction (IDR) methods implemented in Julia/IterativeSolvers package.

TABLE 1. Simulation setups

Cases	N	M	(C_1, C_2)	d_k
Case 1	5000	50	$(8, 1)$	1
Case 2	5000	100	$(15, 3)$	$100\sin(k)^2$
Case 3	10000	100	$(12, 2)$	$5\sin(k)$

Result summary: First, we investigate whether different methods converge within the computational budget for all five initializations. We observe that the multiple shooting, ILQR, and FOTD converge by triggering (7.2) for all three cases in Table 1 and for all initializations and setups. The Schwarz converges for most of cases, but does not converge *within the budget* for Case 2 with $b = 1$. In particular, for this case, there are $2, 4, 5$ out of 5 initializations that the Schwarz does not trigger (7.2) when setting $\mu = 1, 25, 125$, respectively. For ADMM, it converges for all three cases with $\mu = 1$, but does not converge for all three cases with $\mu = 25$ and 125. We do not claim that ADMM diverges with large μ (its KKT is actually below 10^{-3} with > 2000 iterations), but we clearly see that ADMM is not as robust as the Schwarz and FOTD to the parameter μ.

Second, we draw the KKT convergence plots for FOTD in Figure 2. We apply the sparse LU solver for solving QPs and, for Cases 1, 2, and 3, we take $\mu = 1$, 25, and 125 as examples respectively. We emphasize that the other setups of μ of each case have similar convergence behavior (as revealed by



(a) Case 1, $b=1$ (b) Case 1, $b=5$ (c) Case 1, $b=25$

(d) Case 2, $b=1$ (e) Case 2, $b=5$ (f) Case 2, $b=25$

(g) Case 3, $b=1$ (h) Case 3, $b=5$ (i) Case 3, $b=25$

FIGURE 2. Convergence figures. The three plots on each row correspond to the same case in Table 1, while the three plots on each column correspond to the same setup of b. Each plot has 5 lines corresponding to 5 initial iterates. The blue line that is separated from the other four lines corresponds to the initial point $(\boldsymbol{z}^0, \boldsymbol{\lambda}^0) = (\mathbf{0}, \mathbf{0})$. We observe that FOTD converges for all initializations. It exhibits between linear and superlinear convergence locally, and a larger b generally leads to a faster convergence.

Tables 2-4). From Figure 2, we observe that FOTD converges for all initializations with different μ for different cases and exhibits between linear and superlinear convergence locally. Its performance is robust to μ, and a larger b generally leads to a faster convergence (cf. Figures 2g-2i). Our observation is consistent with Theorem 6.4.

Third, we report the KKT residual and running time for all methods. We average the results over the convergent runs among five runs (corresponding to five initializations). Since ADMM converges within the budget for $\mu=1$ only, we report its results under this setup. The results for Cases 1, 2, and 3 are summarized in Tables 2, 3, and 4, respectively. From the tables, we have the following observations. (i) The proposed FOTD and Schwarz outperform other parallel methods such as the multiple shooting, ILQR, and ADMM, among which ADMM has the worst performance. We believe the reasons are two folds. First, the Schwarz [38] and FOTD employ an overlapping decomposition. The overlaps facilitate the information exchange between subproblems, and effectively suppress the

TABLE 2. The KKT residual and running time for all methods implemented on Case 1. For the Schwarz and FOTD, the smallest KKT residual and least running time among different setups of b are highlighted for each setup of μ.

Type	Method		KKT residual (10^{-7})			Time (sec.)		
Centralized	IPOPT		172.686			0.851		
Decomposed	MultiShoot		23.375			7.863		
	ILQR		29.268			7.047		
	ADMM		2603.063			36.125		
			$b=1$	$b=5$	$b=25$	$b=1$	$b=5$	$b=25$
	FOTD (sparse LU)	$\mu=1$	13.324	0.878	**0.0979**	0.474	**0.455**	0.881
		$\mu=25$	4.828	1.618	**0.0980**	0.460	**0.449**	0.885
		$\mu=125$	7.210	0.328	**0.0977**	0.459	**0.453**	0.881
	FOTD (GMRES)	$\mu=1$	4.686	0.672	**0.549**	0.566	**0.537**	1.019
		$\mu=25$	12.841	0.897	**0.132**	0.597	**0.559**	1.026
		$\mu=125$	4.997	**0.251**	0.451	0.621	**0.604**	1.041
	FOTD (IDR)	$\mu=1$	4.051	0.275	**0.106**	0.619	**0.468**	0.875
		$\mu=25$	2.688	0.832	**0.0982**	0.500	**0.467**	0.896
		$\mu=125$	5.206	0.380	**0.0978**	0.536	**0.491**	0.878
	Schwarz	$\mu=1$	0.448	0.298	**0.0418**	**1.969**	2.197	2.009
		$\mu=25$	1.451	0.454	**0.0418**	2.116	2.199	**2.027**
		$\mu=125$	3.467	0.521	**0.0422**	2.173	2.185	**2.160**

TABLE 3. The KKT residual and running time for all methods implemented on Case 2. For the Schwarz and FOTD, the smallest KKT residual and least running time among different setups of b are highlighted for each setup of μ. The dash "-" means the scheme does not trigger (7.2) within the budget.

Type	Method		KKT residual (10^{-7})			Time (sec.)		
Centralized	IPOPT		31.227			1.075		
Decomposed	MultiShoot		15.998			10.126		
	ILQR		26.685			9.334		
	ADMM		11841.583			35.133		
			$b=1$	$b=5$	$b=25$	$b=1$	$b=5$	$b=25$
	FOTD (sparse LU)	$\mu=1$	607.625	117.596	**1.108**	0.708	**0.599**	0.863
		$\mu=25$	364.783	106.214	**0.848**	0.699	**0.601**	0.906
		$\mu=125$	240.339	**0.268**	4.753	0.735	**0.557**	1.026
	FOTD (GMRES)	$\mu=1$	798.660	115.055	**56.258**	1.224	**1.084**	1.503
		$\mu=25$	193.723	**5.581**	12.745	1.265	**1.077**	1.652
		$\mu=125$	738.168	62.946	**33.445**	1.310	**1.029**	1.520
	FOTD (IDR)	$\mu=1$	662.446	37.652	**8.290**	0.828	**0.695**	1.098
		$\mu=25$	310.175	48.570	**7.662**	0.848	**0.724**	1.203
		$\mu=125$	160.98	**9.470**	15.706	0.900	**0.754**	1.245
	Schwarz	$\mu=1$	11.675	14.911	**1.916**	2.427	2.316	**2.055**
		$\mu=25$	14.402	14.770	**1.203**	2.852	2.335	**2.057**
		$\mu=125$	-	22.714	**9.227**	-	2.229	**2.155**

system perturbations brought by the horizon truncation and different choices of μ. Second, as also observed in [38, Figure 6], the ADMM iterates often generate a small stepsize, which is less effective than the stepsize that is selected by the line search. As the augmented Lagrangian method, ADMM also suffers when it is initialized with a poor penalty parameter and/or poor Lagrange multipliers, especially for nonconvex problems [14]. Note that our objective coefficient for the control variables in (7.1a) is $-C_2$ with $C_2 > 0$; thus, (7.1) is nonconvex even if we have a linear system in the constraints.

TABLE 4. The KKT residual and running time for all methods implemented on Case 3. For the Schwarz and FOTD, the smallest KKT residual and least running time among different setups of b are highlighted for each setup of μ.

Type	Method		KKT residual (10^{-7})			Time (sec.)		
Centralized	IPOPT		285.911			1.405		
Decomposed	MultiShoot		5.652			18.571		
	ILQR		19.983			15.732		
	ADMM		28467.487			38.571		
			$b=1$	$b=5$	$b=25$	$b=1$	$b=5$	$b=25$
	FOTD (sparse LU)	$\mu=1$	69.097	159.907	**1.471**	1.067	**1.048**	1.480
		$\mu=25$	23.867	6.988	**0.714**	1.161	**0.942**	1.480
		$\mu=125$	4.944	23.393	**1.906**	1.066	**1.038**	1.513
	FOTD (GMRES)	$\mu=1$	4.293	28.472	**0.0544**	1.382	**1.373**	1.903
		$\mu=25$	47.298	12.676	**0.624**	1.490	**1.284**	1.877
		$\mu=125$	**5.838**	11.084	17.412	1.559	**1.355**	1.897
	FOTD (IDR)	$\mu=1$	144.449	5.971	**3.612**	1.110	**0.989**	1.571
		$\mu=25$	40.627	0.735	**0.532**	1.147	**1.020**	1.557
		$\mu=125$	31.889	29.349	**0.0760**	1.140	**1.062**	1.574
	Schwarz	$\mu=1$	2.363	1.691	**0.0154**	4.914	3.994	**2.686**
		$\mu=25$	1.383	0.134	**0.0166**	4.441	3.485	**2.711**
		$\mu=125$	3.127	0.182	**0.0155**	4.937	3.536	**2.715**

TABLE 5. The KKT residual and running time for all methods implemented on the temperature control problem. For the Schwarz and FOTD, the smallest KKT residual and least running time among different setups of b are highlighted for each setup of μ. For ADMM, the best results among different μ are highlighted.

Type	Method		KKT residual (10^{-7})			Time (sec.)		
Centralized	IPOPT		0.889			16.934		
Decomposed	MultiShoot		3937.071			104.021		
	ILQR		2941.207			89.655		
	IDDP		2619.498			93.832		
	ADMM		$\mu=1$	$\mu=25$	$\mu=125$	$\mu=1$	$\mu=25$	$\mu=125$
			1680.350	11684.271	61455.896	**76.732**	224.930	479.194
			$b=1$	$b=5$	$b=25$	$b=1$	$b=5$	$b=25$
	FOTD (sparse LU)	$\mu=1$	13.621	13.488	**8.050**	16.914	**15.134**	25.531
		$\mu=25$	28.266	8.024	**5.316**	19.973	**18.023**	28.752
		$\mu=125$	11.448	3.112	**2.712**	19.416	**18.356**	28.841
	FOTD (GMRES)	$\mu=1$	17.837	15.228	**2.368**	**16.653**	16.729	21.293
		$\mu=25$	61.917	6.623	**0.573**	21.023	**17.350**	22.830
		$\mu=125$	15.052	2.937	**1.704**	23.336	**17.871**	23.571
	FOTD (IDR)	$\mu=1$	14.934	13.515	**3.998**	**17.106**	17.425	21.393
		$\mu=25$	33.351	**6.954**	8.766	18.459	**16.515**	21.888
		$\mu=125$	28.830	8.553	**7.119**	20.244	**17.318**	22.750
	Schwarz	$\mu=1$	10.995	8.750	**1.303**	23.873	**21.155**	21.724
		$\mu=25$	12.976	4.260	**2.991**	24.503	**22.410**	23.306
		$\mu=125$	5.124	3.501	**1.996**	23.506	**20.431**	22.359

(ii) With three different (but efficient) QP solvers, FOTD performs equally well, although GMRES has slightly longer running time (within 0.3 sec.) than the other two solvers. Thus, different linear system solvers can be employed in FOTD to accelerate its computation. (iii) Between Schwarz and FOTD, the Schwarz tends to attain a smaller KKT residual than FOTD, which is more significant when the overlap size b is as small as 1. For the majority of cases, both methods attain the smallest

KKT residual when b is as large as 25, while attain the largest KKT residual when b is as small as 1. As for the running time, FOTD consistently converges faster than the Schwarz for different choices of μ and b. The running time of FOTD is comparable to that of the centralized solver IPOPT. For the three choices of b, the Schwarz tends to converge faster for a large b than for moderate or small b. This is because that the Schwarz performs less iterations when b is large even if solving each subproblem is also more expensive. On the contrary, FOTD converges faster for a moderate b, which reveals the trade-off between the total number of iterations and the computation cost of a single iteration. Overall, our experiments demonstrate that both FOTD and Schwarz are superior parallel methods and robust to the parameter μ. FOTD is more efficient than the Schwarz, and is as efficient as the popular solver IPOPT. However, as the parallel method, FOTD (and Schwarz) offers more flexibility to the computing environments and can be applied when a single high-speed processor is not accessible.

Thin plate temperature control: We apply FOTD on a thin plate temperature control problem studied in [38]. We refer to [8, 9, 33] for the physical background of the problem. In particular, we let $k \in [0,1]$ be the continuous time index and $\boldsymbol{\omega} \in \Omega = [0,1] \times [0,1]$ be the position in the domain Ω. Then, we consider the following problem

$$\min_{x,u} \int_0^1 \int_{\boldsymbol{\omega}\in\Omega} \left\{ (\boldsymbol{x}(\boldsymbol{\omega},k) - d(\boldsymbol{\omega},k))^2 + u(\boldsymbol{\omega},k)^2 \right\} d\boldsymbol{\omega}\, dk, \tag{7.3a}$$

$$\text{s.t. } \frac{\partial x(\boldsymbol{\omega},k)}{\partial k} = \nabla^2 x(\boldsymbol{\omega},k) + u(\boldsymbol{\omega},k) + \frac{2h_c}{\kappa_c t_c}(T_c - x(\boldsymbol{\omega},k)) + \frac{2\epsilon_c\sigma_c}{\kappa_c t_c}(T_c^4 - x(\boldsymbol{\omega},k)^4), \quad \forall k \in [0,1], \forall \boldsymbol{\omega} \in \Omega, \tag{7.3b}$$

$$x(\boldsymbol{\omega},k) = 0, \quad \forall (\boldsymbol{\omega},k) \in \Omega \times \{0\} \text{ or } \partial\Omega \times [0,1], \tag{7.3c}$$

where ∇^2 is the Laplace operator, that is, $\nabla^2 x = \frac{\partial^2 x}{\partial \boldsymbol{\omega}_1^2} + \frac{\partial^2 x}{\partial \boldsymbol{\omega}_2^2}$, and $d(\boldsymbol{\omega},k)$ in the objective (7.3a) is the prespecified desired temperature. Here, the PDE constraints in (7.3b) are governed by a controlled heat equation (i.e., the first two terms) with extra convection and radiation terms (i.e., the third and fourth terms). All the symbols with a subscript "c" are the prespecified constants. In particular, h_c is the convection coefficient; κ_c is the thermal conductivity; ϵ_c is the emissivity coefficient; σ_c is the Stefan-Boltzmann constant; T_c is the ambient temperature; and t_c is the plate thickness. We unify the coefficients of heat equation for simplicity. The initial and boundary conditions are in (7.3c).

In our implementation, we discretize Ω by a 4×4 mesh grid, i.e. 2×2 mesh grid in the interior. The temporal horizon is decomposed by 5000 evenly spaced knots with 50 knots for each subproblem. The setups of all methods are as before, and we follow [33] to set up the problem parameters. In particular, we let $h_c = 1$, $\kappa_c = 400$, $\epsilon_c = 0.5$, $\sigma_c = 5.67 \times 10^{-8}$, $T_c = 300$, $t_c = 0.01$, and let $d(\boldsymbol{\omega},k) = \sin(k)$. The KKT residual and running time of the methods are summarized in Table 5. From the table, we again observe that the Schwarz and FOTD outperform other four parallel methods. The Schwarz attains smaller KKT residual than FOTD, while FOTD converges faster than the Schwarz. Overall, our experiment shows the superiority of the overlapping decomposition-based methods.

8. Conclusion This paper proposes a fast overlapping temporal decomposition (FOTD) procedure for solving long-horizon NLDPs in (1.1). FOTD relies on the sequential quadratic programming (SQP) and incorporates SQP with OTD technique. We establish global convergence and uniform, local linear convergence for FOTD. The local result matches [38], while FOTD requires fewer computations in each iteration (cf. Theorem 6.2).

Considering the improvement of the performance of the Schwarz scheme over ADMM [38], and the relation between FOTD and the Schwarz, we believe the extension of FOTD is worth studying. One of the drawbacks of FOTD is the separation of modifying the Hessian matrix and solving the linear-quadratic subproblems (steps 1 and 2 in section 3), which leads to two separate factorizations

for the subproblem matrices. Such a drawback does not enlarge the flops order of a single processor, but indeed results in a suboptimal flop multiplier. A more desirable algorithm should perform the Hessian modification and solve the subproblem in a single machine jointly, with one factorization, such as parallel quasi-Newton scheme [11]. Further, we can embed OTD into more advanced SQP frameworks, such as the trust region-SQP. We can also replace the line search step by the filter step, and study the behavior of the approximate direction obtained by OTD on the filter step.

In addition, FOTD can be applied on graph-structured problems, seeing that a similar exponential decay of sensitivity for graph-structured problems was established in [51]. Finally, a reasonable conjecture for the improved performance of the Schwarz scheme and FOTD over ADMM (and other parallel methods) is the lack of information exchange among subproblems in ADMM. No overlaps are adopted in ADMM. Thus, whether we can embed OTD into ADMM to improve the performance of ADMM, and whether the OTD-based ADMM exhibits a similar convergence rate as FOTD are interesting future research directions.

Acknowledgments. This material was based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR) under Contract DE-AC02-06CH11347 and by NSF through award CNS-1545046.

Appendix A: Proofs of results in section 2

A.1. Proof of Theorem 2.1 We start by writing the KKT conditions of Problem (1.1) and Problem (2.2). For $k \in [N-1]$, we let $A_k(\boldsymbol{z}_k) = \nabla^T_{\boldsymbol{x}_k} f_k(\boldsymbol{z}_k)$, $B_k(\boldsymbol{z}_k) = \nabla^T_{\boldsymbol{u}_k} f_k(\boldsymbol{z}_k)$ be the Jacobian matrices. Then the Lagrange function of (1.1) is

$$\mathcal{L}(\boldsymbol{z}, \boldsymbol{\lambda}) = \sum_{k=0}^{N-1} \{g_k(\boldsymbol{z}_k) + \boldsymbol{\lambda}_k^T \boldsymbol{x}_k - \boldsymbol{\lambda}_{k+1}^T f_k(\boldsymbol{z}_k)\} + \{g_N(\boldsymbol{x}_N) + \boldsymbol{\lambda}_N^T \boldsymbol{x}_N\} - \boldsymbol{\lambda}_0^T \bar{\boldsymbol{x}}_0. \tag{A.1}$$

Thus, the KKT conditions of (1.1) are

$$\begin{aligned}
\nabla_{\boldsymbol{x}_k} g_k(\boldsymbol{z}_k) + \boldsymbol{\lambda}_k - A_k^T(\boldsymbol{z}_k)\boldsymbol{\lambda}_{k+1} = \boldsymbol{0}, \quad & \forall k \in [N-1], && \text{(A.2a)}\\
\nabla_{\boldsymbol{u}_k} g_k(\boldsymbol{z}_k) - B_k^T(\boldsymbol{z}_k)\boldsymbol{\lambda}_{k+1} = \boldsymbol{0}, \quad & \forall k \in [N-1], && \text{(A.2b)}\\
\nabla_{\boldsymbol{x}_N} g_N(\boldsymbol{z}_N) + \boldsymbol{\lambda}_N = \boldsymbol{0}, \quad & && \text{(A.2c)}\\
\boldsymbol{x}_{k+1} - f_k(\boldsymbol{z}_k) = \boldsymbol{0}, \quad & \forall k \in [N-1], && \text{(A.2d)}\\
\boldsymbol{x}_0 - \bar{\boldsymbol{x}}_0 = \boldsymbol{0}. \quad & && \text{(A.2e)}
\end{aligned}$$

Similarly, the KKT conditions of $\mathcal{P}_\mu^i(\boldsymbol{d}_i)$ in (2.2) are

$$\begin{aligned}
\nabla_{\boldsymbol{x}_k} g_k(\tilde{\boldsymbol{z}}_{i,k}) + \tilde{\boldsymbol{\lambda}}_{i,k} - A_k^T(\tilde{\boldsymbol{z}}_{i,k})\tilde{\boldsymbol{\lambda}}_{i,k+1} = \boldsymbol{0}, \quad & \forall k \in [m_1, m_2), && \text{(A.3a)}\\
\nabla_{\boldsymbol{u}_k} g_k(\tilde{\boldsymbol{z}}_{i,k}) - B_k^T(\tilde{\boldsymbol{z}}_{i,k})\tilde{\boldsymbol{\lambda}}_{i,k+1} = \boldsymbol{0}, \quad & \forall k \in [m_1, m_2), && \text{(A.3b)}\\
\nabla_{\boldsymbol{x}_{m_2}} g_{m_2}(\tilde{\boldsymbol{x}}_{i,m_2}, \bar{\boldsymbol{u}}_{m_2}) + \tilde{\boldsymbol{\lambda}}_{i,m_2} - A_{m_2}^T(\tilde{\boldsymbol{x}}_{i,m_2}, \bar{\boldsymbol{u}}_{m_2})\bar{\boldsymbol{\lambda}}_{m_2+1} & && \\
+\mu(\tilde{\boldsymbol{x}}_{i,m_2} - \bar{\boldsymbol{x}}_{m_2}) = \boldsymbol{0}, \quad & && \text{(A.3c)}\\
\tilde{\boldsymbol{x}}_{i,k+1} - f_k(\tilde{\boldsymbol{z}}_{i,k}) = \boldsymbol{0}, \quad & \forall k \in [m_1, m_2), && \text{(A.3d)}\\
\tilde{\boldsymbol{x}}_{i,m_1} - \bar{\boldsymbol{x}}_{m_1} = \boldsymbol{0}. \quad & && \text{(A.3e)}
\end{aligned}$$

(i). By Definition 2.1, $\mathcal{D}_i(\boldsymbol{z}^\star, \boldsymbol{\lambda}^\star) = (\boldsymbol{x}^\star_{m_1:m_2}, \boldsymbol{u}^\star_{m_1:m_2-1}, \boldsymbol{\lambda}^\star_{m_1:m_2})$. Letting $\boldsymbol{d}_i = \boldsymbol{d}_i^\star$ in (A.3c), (A.3e), we see (A.3) is a subsystem of (A.2) so that $\mathcal{D}_i(\boldsymbol{z}^\star, \boldsymbol{\lambda}^\star)$ satisfies (A.3). Thus, the statement holds. **(ii)**. Suppose $(\tilde{\boldsymbol{z}}_i, \tilde{\boldsymbol{\lambda}}_i) = (\tilde{\boldsymbol{z}}_i^\star(\boldsymbol{d}_i), \tilde{\boldsymbol{\lambda}}_i^\star(\boldsymbol{d}_i))$ satisfies conditions (A.3). Since $[n_i, n_{i+1}) \subseteq [m_1, m_2)$, we know from (A.3a), (A.3b), (A.3d) that $(\tilde{\boldsymbol{z}}_{i,n_i:n_{i+1}-1}, \tilde{\boldsymbol{\lambda}}_{i,n_i:n_{i+1}-1})$ satisfies

$$\begin{aligned}
\nabla_{\boldsymbol{x}_k} g_k(\tilde{\boldsymbol{z}}_{i,k}) + \tilde{\boldsymbol{\lambda}}_{i,k} - A_k^T(\tilde{\boldsymbol{z}}_{i,k})\tilde{\boldsymbol{\lambda}}_{i,k+1} = \boldsymbol{0}, \quad & \forall k \in [n_i, n_{i+1}), && \text{(A.4a)}\\
\nabla_{\boldsymbol{u}_k} g_k(\tilde{\boldsymbol{z}}_{i,k}) - B_k^T(\tilde{\boldsymbol{z}}_{i,k})\tilde{\boldsymbol{\lambda}}_{i,k+1} = \boldsymbol{0}, \quad & \forall k \in [n_i, n_{i+1}), && \text{(A.4b)}\\
\tilde{\boldsymbol{x}}_{i,k+1} - f_k(\tilde{\boldsymbol{z}}_{i,k}) = \boldsymbol{0}, \quad & \forall k \in [n_i, n_{i+1}), && \text{(A.4c)}
\end{aligned}$$

which is a subset of (A.2a), (A.2b), (A.2d). We consider the composed point $(\boldsymbol{z},\boldsymbol{\lambda})=\mathcal{C}(\{(\tilde{\boldsymbol{z}}_i,\tilde{\boldsymbol{\lambda}}_i)\}_i)$. By Definition 2.1, we know

$$\begin{aligned}&\boldsymbol{z}_k=\tilde{\boldsymbol{z}}_{i,k},\quad \text{if } k\in[n_i,n_{i+1}) \text{ for some } i\in[M-1], \qquad \boldsymbol{z}_N=\tilde{\boldsymbol{x}}_{M-1,N},\\ &\boldsymbol{\lambda}_k=\tilde{\boldsymbol{\lambda}}_{i,k},\quad \text{if } k\in[n_i,n_{i+1}) \text{ for some } i\in[M-1], \qquad \boldsymbol{\lambda}_N=\tilde{\boldsymbol{\lambda}}_{M-1,N}.\end{aligned} \tag{A.5}$$

Thus, (A.2c) is implied by (A.3c) for $i=M-1$, and (A.2e) is implied by (A.3e) for $i=0$. For (A.2b), we use the decomposition $[N-1]=\cup_{i=0}^{M-1}[n_i,n_{i+1})$. For any $k\in[N-1]$, we have two cases.
(a). $k\neq n_{i+1}-1$, $\forall i\in[M-2]$. Then, $k\in[n_i,n_{i+1}-1)$ for some $i\in[M-2]$, or $k\in[n_{M-1},n_M)$. Thus, $k+1\in[n_i+1,n_{i+1})$ for $i\in[M-2]$ or $k+1\in[n_{M-1}+1,N]$. By (A.5), we know for both cases that $\boldsymbol{z}_k$ and $\boldsymbol{\lambda}_{k+1}$ are from the same subproblem, i.e. $\boldsymbol{z}_k=\tilde{\boldsymbol{z}}_{i,k}$ and $\boldsymbol{\lambda}_{k+1}=\tilde{\boldsymbol{\lambda}}_{i,k+1}$. Thus, (A.2b) is implied by (A.4b).
(b). $k=n_{i+1}-1$ for $i\in[M-2]$. Then, $k+1=n_{i+1}$ and thus $\boldsymbol{\lambda}_{k+1}=\tilde{\boldsymbol{\lambda}}_{i+1,k+1}$. Comparing (A.2b) with (A.4b), we know that

$$\begin{aligned}\nabla_{\boldsymbol{u}_k}g_k(\boldsymbol{z}_k)-B_k^T(\boldsymbol{z}_k)\boldsymbol{\lambda}_{k+1}=\boldsymbol{0}&\Longleftrightarrow \nabla_{\boldsymbol{u}_k}g_k(\tilde{\boldsymbol{z}}_{i,k})-B_k^T(\tilde{\boldsymbol{z}}_{i,k})\tilde{\boldsymbol{\lambda}}_{i+1,k+1}=\boldsymbol{0}\\ &\overset{(A.4b)}{\Longleftrightarrow} B_k^T(\tilde{\boldsymbol{z}}_{i,k})\tilde{\boldsymbol{\lambda}}_{i+1,k+1}=B_k^T(\tilde{\boldsymbol{z}}_{i,k})\tilde{\boldsymbol{\lambda}}_{i,k+1}.\end{aligned} \tag{A.6}$$

Following the same derivation, we consider (A.2a) and can easily get

$$\nabla_{\boldsymbol{x}_k}g_k(\boldsymbol{z}_k)+\boldsymbol{\lambda}_k-A_k^T(\boldsymbol{z}_k)\boldsymbol{\lambda}_{k+1}=\boldsymbol{0}\overset{(A.4a)}{\Longleftrightarrow}A_k^T(\tilde{\boldsymbol{z}}_{i,k})\tilde{\boldsymbol{\lambda}}_{i+1,k+1}=A_k^T(\tilde{\boldsymbol{z}}_{i,k})\tilde{\boldsymbol{\lambda}}_{i,k+1}, \tag{A.7}$$

if $k=n_{i+1}-1$ for $i\in[M-2]$. Finally, we consider (A.2d), where we have two cases.
(a). $k\neq n_{i+1}-1$, $\forall i\in[M-2]$. As before, $\boldsymbol{z}_k$ and $\boldsymbol{x}_{k+1}$ are from the same subproblem, so that (A.2d) is implied by (A.4c).
(b). $k=n_{i+1}-1$ for $i\in[M-2]$. Then, $\boldsymbol{x}_{k+1}=\tilde{\boldsymbol{x}}_{i+1,k+1}$. Comparing (A.2d) with (A.4c), we know

$$\boldsymbol{x}_{k+1}=f_k(\boldsymbol{z}_k)\Longleftrightarrow\tilde{\boldsymbol{x}}_{i+1,k+1}=f_k(\tilde{\boldsymbol{z}}_{i,k})\overset{(A.4c)}{\Longleftrightarrow}\tilde{\boldsymbol{x}}_{i+1,k+1}=\tilde{\boldsymbol{x}}_{i,k+1}. \tag{A.8}$$

Combining (A.6), (A.7), and (A.8), we complete the proof.

Appendix B: Proofs of results in section 4

B.1. Proof of Lemma 4.1 We suppress the iteration index τ since the result holds uniformly over τ. By Assumption 4.2, we know that for any integer k, there exists an integer $t_k\in[1,t]$ such that $\Xi_{k,t_k}\Xi_{k,t_k}^T\succeq\gamma_C I$. Let us define the knots recursively by $k_{j+1}=k_j+t_{k_j}$, $j=0,1,\ldots,J-1$ with $k_0=0$. We suppose J is large enough such that $k_J>N-t$. Thus, we derive a horizon decomposition $[N-1]=\cup_{j=0}^{J-1}[k_j,k_{j+1}-1]\cup[k_J,N-1]$. Since $G=\nabla_{\boldsymbol{z}}^T f$, we can apply the definition of f in (3.1) and write GG^T explicitly. We have

$$GG^T=\begin{pmatrix} I & -A_0^T & & & \\ -A_0 & I+A_0A_0^T+B_0B_0^T & \ddots & & \\ & \ddots & \ddots & -A_{N-1}^T & \\ & & -A_{N-1} & I+A_{N-1}A_{N-1}^T+B_{N-1}B_{N-1}^T & \end{pmatrix}. \tag{B.1}$$

We let $k_j'=k_{j+1}-1=k_j+t_{k_j}-1$, and define matrices $\{\mathcal{T}_j\}_{j\in[J]}$ corresponding to each interval $[k_j,k_j']$ as ($\mathcal{T}_J$ is defined similarly, except that the bottom corner block is $I+A_{N-1}A_{N-1}^T+B_{N-1}B_{N-1}^T$)

$$\mathcal{T}_j=\begin{pmatrix} I & -A_{k_j}^T & & & \\ -A_{k_j} & I+A_{k_j}A_{k_j}^T+B_{k_j}B_{k_j}^T & \ddots & & \\ & \ddots & \ddots & -A_{k_j'}^T & \\ & & -A_{k_j'} & A_{k_j'}A_{k_j'}^T+B_{k_j'}B_{k_j'}^T & \end{pmatrix},\quad \forall j\in[J-1].$$

By the expression in (B.1), it suffices to show that each $\mathcal{T}_j$ is lower bounded away from zero. Then, we have $\lambda_{\min}(GG^T) \geq \min_{j\in[J]} \lambda_{\min}(\mathcal{T}_j)$. Let us consider $\mathcal{T}_J$ first. We let $A_m^n = A_m A_{m-1} \cdots A_n$ and define a matrix $\mathcal{T}_J^{1/2}$ as

$$\begin{aligned}\mathcal{T}_J^{1/2} &:= \begin{pmatrix} I & & & & & \\ -A_{k_J} & I & & & -B_{k_J} & \\ & \ddots & \ddots & & & \ddots \\ & & & I & & -B_{N-2} \\ & & & -A_{N-1} & I & -B_{N-1} \end{pmatrix} \\ &= \begin{pmatrix} I & & & \\ -A_{k_J} & I & & \\ & \ddots & & \\ & & I & \end{pmatrix} \times \cdots \times \begin{pmatrix} I & & & \\ & I & & \\ & & \ddots & \\ & & & I \\ & & -A_{N-1} & I \end{pmatrix} \times \begin{pmatrix} I & & & & & \\ & I & & & -B_{k_J} & \\ & \ddots & & \vdots & & \ddots \\ & & I & -A_{N-2}^{k_J+1} B_{k_J} & & -B_{N-2} \\ & & & I & -A_{N-1}^{k_J+1} B_{k_J} & -A_{N-1}B_{N-2} \ -B_{N-1} \end{pmatrix} \\ &=: P_{k_J} \cdots P_{N-1} Q_J.\end{aligned}$$

Then, we use the fact that $Q_J Q_J^T \succeq I$ and have

$$\mathcal{T}_J = \mathcal{T}_J^{1/2} (\mathcal{T}_J^{1/2})^T = P_{k_J} \cdots P_{N-1} Q_J Q_J^T P_{N-1}^T \cdots P_{k_J}^T \succeq P_{k_J} \cdots P_{N-1} P_{N-1}^T \cdots P_{k_J}^T.$$

Since $\|A_k\| \leq \Upsilon_{upper}$ for $k \in [k_J, N-1]$, we know $\|P_k^{-1}\| \leq 1 + \Upsilon_{upper}$. Thus, $P_k P_k^T \succeq 1/(1+\Upsilon_{upper})^2 I$. By the above display and using $N - k_J \leq t$, we have $\lambda_{\min}(\mathcal{T}_J) \succeq 1/(1+\Upsilon_{upper})^{2t}$. We then consider $\mathcal{T}_j$ for $j \in [J-1]$ similarly. By the same matrix multiplication, we have

$$\mathcal{T}_j^{1/2} := \begin{pmatrix} I & & & & & \\ -A_{k_j} & I & & & -B_{k_j} & \\ & \ddots & \ddots & & & \ddots \\ & & & I & & -B_{k'_j-1} \\ & & & -A_{k'_j} & & -B_{k'_j} \end{pmatrix} = P_{k_j} \cdots P_{k'_j} Q_j,$$

where (slightly different from Q_J)

$$Q_j = \begin{pmatrix} I & & & & & \\ & I & & -B_{k_j} & & \\ & & \ddots & \vdots & \ddots & \\ & & & I \ -A_{k'_j-1}^{k_j+1} B_{k_j} & & -B_{k'_j-1} \\ & & & -A_{k'_j}^{k_j+1} B_{k_j} & & -A_{k'_j} B_{k'_j-1} \ -B_{k'_j} \end{pmatrix} =: \begin{pmatrix} I & Q_{j,1} \\ \mathbf{0} & Q_{j,2} \end{pmatrix}.$$

Since $Q_{j,2}$ is just the controllability matrix $\Xi_{k_j, t_{k_j}}$ (except that the components are in a reverse order), Assumption 4.2 implies that $Q_{j,2} Q_{j,2}^T \succeq \gamma_C I$. Furthermore, since $\max\{\|A_k\|, \|B_k\|\} \leq \Upsilon_{upper}$, $\forall k \in [k_j, k'_j - 1]$, we get

$$\max\{\|Q_{j,2}\|, \|Q_{j,1}\|\} \leq \Upsilon_{upper} + \Upsilon_{upper}^2 + \cdots \Upsilon_{upper}^{k'_j - k_j + 1} \leq \sum_{i=1}^{t} \Upsilon_{upper}^i \leq \frac{\Upsilon_{upper}^{t+1}}{\Upsilon_{upper} - 1},$$

where the second inequality is due to $t_{k_j} \leq t$. Finally, noting that

$$Q_j Q_j^T = \begin{pmatrix} I + Q_{j,1} Q_{j,1}^T & Q_{j,1} Q_{j,2}^T \\ Q_{j,2} Q_{j,1}^T & Q_{j,2} Q_{j,2}^T \end{pmatrix},$$

we apply the algebra result in [34, Lemma 4.8(ii)] and obtain

$$Q_j Q_j^T \succeq \left(\frac{\gamma_C}{\gamma_C + \frac{\Upsilon_{upper}^{t+1}}{\Upsilon_{upper} - 1}} \right)^2 \cdot \min\{1, \gamma_C\} \cdot I =: \gamma_Q I.$$

Thus, using $P_k P_k^T \succeq 1/(1+\Upsilon_{upper})^2 I$, we obtain

$$\mathcal{T}_j = \mathcal{T}_j^{1/2}(\mathcal{T}_j^{1/2})^T = P_{k_j}\cdots P_{k'_j} Q_j Q_j^T P_{k'_j}^T \cdots P_{k_j}^T \succeq \frac{\gamma_Q}{(1+\Upsilon_{upper})^{2t}} I.$$

Since $\gamma_Q < 1$, we let $\gamma_G = \gamma_Q/(1+\Upsilon_{upper})^{2t}$ and have $\lambda_{\min}(\mathcal{T}_j) \succeq \gamma_G$, $\forall j \in [J]$. This finishes the proof.

B.2. Proof of Lemma 4.2 We adapt the proof of [35, Lemma 1 and Theorem 1]. We suppress the iteration index τ. The reduced Hessian of $\mathcal{LP}_\mu^i$ is independent from $\boldsymbol{d}_i$, which only affects linear terms. From (3.9b)-(3.9c), we know that the Jacobian matrix of the constraints in $\mathcal{LP}_\mu^i$ has full row rank. Thus, it suffices to show that the reduced Hessian of $\mathcal{LP}_\mu^i$ is lower bounded by $\gamma_{RH} I$. Let

$$\hat{H}^{(i)} = \operatorname{diag}(\hat{H}_{m_1}, \hat{H}_{m_1+1}, \ldots, \hat{H}_{m_2-1}, \hat{Q}_{m_2} + \mu I)$$

be the Hessian of $\mathcal{LP}_\mu^i$. We only need to show that $\tilde{\boldsymbol{w}}_i^T \hat{H}^{(i)} \tilde{\boldsymbol{w}}_i \geq \gamma_{RH} \|\tilde{\boldsymbol{w}}_i\|^2$ for any $\tilde{\boldsymbol{w}}_i = (\tilde{\boldsymbol{p}}_i, \tilde{\boldsymbol{q}}_i) \neq \boldsymbol{0}$ satisfying

$$\tilde{\boldsymbol{p}}_{i,k+1} = A_k \tilde{\boldsymbol{p}}_{i,k} + B_k \tilde{\boldsymbol{q}}_{i,k}, \quad k \in [m_1, m_2), \tag{B.2a}$$
$$\tilde{\boldsymbol{p}}_{i,m_1} = \boldsymbol{0}. \tag{B.2b}$$

We take the last subproblem (i.e., $i = M-1$) as an example to illustrate the proof idea. For $i = M-1$, we have $\mu = 0$. For any $\tilde{\boldsymbol{w}}_{M-1} = (\tilde{\boldsymbol{p}}_{M-1}, \tilde{\boldsymbol{q}}_{M-1}) \neq \boldsymbol{0}$ satisfying (B.2), we extend $\tilde{\boldsymbol{w}}_{M-1}$ forward by filling with $\boldsymbol{0}$ to get a full-horizon vector $\boldsymbol{w}$. That is $\boldsymbol{w} = (\boldsymbol{0}; \tilde{\boldsymbol{w}}_{M-1})$. We can verify that $\boldsymbol{w} = (\boldsymbol{p}, \boldsymbol{q}) \in \{\boldsymbol{w} : G\boldsymbol{w} = \boldsymbol{0}, \boldsymbol{w} \neq \boldsymbol{0}\}$. Therefore,

$$\tilde{\boldsymbol{w}}_{M-1}^T \hat{H}^{(M-1)} \tilde{\boldsymbol{w}}_{M-1} = \boldsymbol{w}^T \hat{H} \boldsymbol{w} \geq \gamma_{RH} \|\boldsymbol{w}\|^2 = \gamma_{RH} \|\tilde{\boldsymbol{w}}_{M-1}\|^2,$$

where the first and last equalities are due to the construction of $\boldsymbol{w}$, and the middle inequality is due to Assumption 4.1. For $i \in [M-2]$, we consider the following two cases.

Case 1: $m_2 \geq N - t$. Given $\tilde{\boldsymbol{w}}_i = (\tilde{\boldsymbol{p}}_i, \tilde{\boldsymbol{q}}_i) \neq \boldsymbol{0}$ satisfying (B.2), we can still extend it forward by filling with $\boldsymbol{0}$. However, extending backward with $\boldsymbol{0}$ will make the full vector $\boldsymbol{w}$ outside the space $\{\boldsymbol{w} : G\boldsymbol{w} = \boldsymbol{0}\}$. Instead, we construct the following extension. We let $\boldsymbol{q}_{m_2:N-1} = \boldsymbol{0}$, $\boldsymbol{p}_{k+1} = A_k \boldsymbol{p}_k$ for $k = [m_2, N)$. Thus,

$$\boldsymbol{w} = (\boldsymbol{0}; \tilde{\boldsymbol{w}}_i; \boldsymbol{0}; \boldsymbol{p}_{m_2+1}; \boldsymbol{0}; \boldsymbol{p}_{m_2+2} \ldots; \boldsymbol{0}; \boldsymbol{p}_N) \in \{\boldsymbol{w} : G\boldsymbol{w} = \boldsymbol{0}, \boldsymbol{w} \neq \boldsymbol{0}\}. \tag{B.3}$$

Moreover, by Assumption 4.3,

$$\begin{aligned}\|\boldsymbol{p}_{m_2+1:N}\|^2 = \sum_{k=m_2+1}^{N} \|\boldsymbol{p}_k\|^2 &\leq \sum_{j=1}^{N-m_2} \Upsilon_{upper}^{2j} \|\boldsymbol{p}_{m_2}\|^2 \\ &= \frac{\Upsilon_{upper}^2 (\Upsilon_{upper}^{2(N-m_2)} - 1)}{\Upsilon_{upper}^2 - 1} \|\boldsymbol{p}_{m_2}\|^2 \leq \frac{\Upsilon_{upper}^{2t+2} - \Upsilon_{upper}^2}{\Upsilon_{upper}^2 - 1} \|\boldsymbol{p}_{m_2}\|^2. \end{aligned} \tag{B.4}$$

The last inequality uses $N - m_2 \leq t$. Further,

$$\begin{aligned}\boldsymbol{w}^T \hat{H} \boldsymbol{w} &\overset{(B.3)}{=} \tilde{\boldsymbol{w}}_i^T \hat{H}^{(i)} \tilde{\boldsymbol{w}}_i - \mu \|\boldsymbol{p}_{m_2}\|^2 + \sum_{k=m_2+1}^{N} \boldsymbol{p}_k^T \hat{Q}_k \boldsymbol{p}_k \leq \tilde{\boldsymbol{w}}_i^T \hat{H}^{(i)} \tilde{\boldsymbol{w}}_i - \mu \|\boldsymbol{p}_{m_2}\|^2 + \Upsilon_{upper} \|\boldsymbol{p}_{m_2+1:N}\|^2 \\ &\overset{(B.4)}{\leq} \tilde{\boldsymbol{w}}_i^T \hat{H}^{(i)} \tilde{\boldsymbol{w}}_i - \left(\mu - \frac{\Upsilon_{upper}(\Upsilon_{upper}^{2t+2} - \Upsilon_{upper}^2)}{\Upsilon_{upper}^2 - 1}\right) \|\boldsymbol{p}_{m_2}\|^2, \end{aligned} \tag{B.5}$$

where the second inequality is due to Assumption 4.3. By Assumption 4.1, $\boldsymbol{w}^T\hat{H}\boldsymbol{w} \geq \gamma_{RH}\|\boldsymbol{w}\|^2 \geq \gamma_{RH}\|\tilde{\boldsymbol{w}}_i\|^2$. Combining with (B.5), we see that $\tilde{\boldsymbol{w}}_i^T\hat{H}^i\tilde{\boldsymbol{w}}_i \geq \gamma_{RH}\|\tilde{\boldsymbol{w}}_i\|^2$ provided

$$\mu \geq \frac{\Upsilon_{upper}(\Upsilon_{upper}^{2t+2} - \Upsilon_{upper}^2)}{\Upsilon_{upper}^2 - 1}. \tag{B.6}$$

Case 2: $m_2 < N - t$. In this case, using the construction of $\boldsymbol{w}$ in (B.3) will result in a bound on μ that grows exponentially in N. Instead, we make use of controllability in Assumption 4.2. In particular, we still let $(\boldsymbol{p}_{0:m_1-1}, \boldsymbol{q}_{0:m_1-1}) = (\boldsymbol{0}, \boldsymbol{0})$ and $\boldsymbol{q}_{m_2} = \boldsymbol{0}$. Then $\boldsymbol{p}_{m_2+1} = A_{m_2}\boldsymbol{p}_{m_2}$. Let $l = m_2 + 1$ and let $(\boldsymbol{p}_k, \boldsymbol{q}_k) = (\boldsymbol{0}, \boldsymbol{0})$, $\forall k \geq l + t_l$. We now show how to evolve from $\boldsymbol{p}_l$ to $\boldsymbol{p}_{l+t_l}$. Applying (B.2a) recursively, we have

$$\boldsymbol{p}_{l+j} = \left(\prod_{h=1}^{j} A_{l+h-1}\right)\boldsymbol{p}_l + \Xi_{l,j}\begin{pmatrix}\boldsymbol{q}_{l+j-1}\\ \vdots \\ \boldsymbol{q}_l\end{pmatrix}, \qquad \forall j \geq 1. \tag{B.7}$$

Letting $j = t_l$ in (B.7), we see that if we specify the control sequence $\boldsymbol{q}_{l+t_l-1:l}$ by

$$\boldsymbol{q}_{l+t_l-1:l} = -\Xi_{l,t_l}^T(\Xi_{l,t_l}\Xi_{l,t_l}^T)^{-1}(\prod_{h=1}^{t_l} A_{l+h-1})\boldsymbol{p}_l, \tag{B.8}$$

and generate $\boldsymbol{p}_{l+j}$ as (B.2a), then we have $\boldsymbol{p}_{l+t_l} = \boldsymbol{0}$. Thus, $\boldsymbol{w} = (\boldsymbol{p}, \boldsymbol{q}) \in \{\boldsymbol{w} : G\boldsymbol{w} = \boldsymbol{0}, \boldsymbol{w} \neq \boldsymbol{0}\}$. Moreover, by Assumptions 4.2 and 4.3, we obtain from (B.8) that

$$\|\boldsymbol{q}_{l+t_l-1:l}\| \leq \|\Xi_{l,t_l}^T(\Xi_{l,t_l}\Xi_{l,t_l}^T)^{-1}\| \cdot \Upsilon_{upper}^{t_l}\|\boldsymbol{p}_l\| \leq \frac{\Upsilon_{upper}^{t_l+1}}{\sqrt{\gamma_C}}\|\boldsymbol{p}_{m_2}\| \leq \frac{\Upsilon_{upper}^{t+1}}{\sqrt{\gamma_C}}\|\boldsymbol{p}_{m_2}\|. \tag{B.9}$$

The second inequality uses the fact that $\|A^T(AA^T)^{-1}\| = \|\Sigma^{-1}\|$ for any full row rank matrix A, and $U\Sigma V^T$ is its singular value decomposition. The last inequality is due to $t_l \leq t$ by Assumption 4.2. Furthermore, for any $1 \leq j \leq t_l - 1$,

$$\begin{aligned}\|\boldsymbol{p}_{l+j}\| &\overset{(B.7)}{\leq} \Upsilon_{upper}^j\|\boldsymbol{p}_l\| + \|\Xi_{l,j}\|\|\boldsymbol{q}_{l+j-1:l}\| \leq \Upsilon_{upper}^j\|\boldsymbol{p}_l\| + \left(\sum_{h=1}^{j}\Upsilon_{upper}^h\right)\|\boldsymbol{q}_{l+t_l-1:l}\| \\ &\overset{(B.9)}{\leq} \Upsilon_{upper}^{j+1}\|\boldsymbol{p}_{m_2}\| + \frac{\Upsilon_{upper}^{j+1} - \Upsilon_{upper}}{\Upsilon_{upper} - 1} \cdot \frac{\Upsilon_{upper}^{t+1}}{\sqrt{\gamma_C}}\|\boldsymbol{p}_{m_2}\|,\end{aligned} \tag{B.10}$$

where the second inequality is due to $\|\Xi_{l,j}\| \leq \sum_{h=1}^{j}\Upsilon_{upper}^h$ by the definition of $\Xi_{l,j}$ in Assumption 4.2. Without loss of generality, we suppose $\Upsilon_{upper}/2 \geq 1 \geq \gamma_C$, then

$$\Upsilon_{upper} \leq \frac{\Upsilon_{upper}^{t+2}}{(\Upsilon_{upper} - 1)\sqrt{\gamma_C}} \leq \frac{2\Upsilon_{upper}^{t+1}}{\sqrt{\gamma_C}}. \tag{B.11}$$

Thus, (B.10) can be further simplified as

$$\begin{aligned}\|\boldsymbol{p}_{l+j}\| &\overset{(B.10)}{\leq} \Upsilon_{upper}^{j+1}\|\boldsymbol{p}_{m_2}\| + \frac{\Upsilon_{upper}^{j+1}}{\Upsilon_{upper} - 1} \cdot \frac{\Upsilon_{upper}^{t+1}}{\sqrt{\gamma_C}}\|\boldsymbol{p}_{m_2}\| \\ &\overset{(B.11)}{\leq} \frac{2\Upsilon_{upper}^{j+1}}{\Upsilon_{upper} - 1} \cdot \frac{\Upsilon_{upper}^{t+1}}{\sqrt{\gamma_C}}\|\boldsymbol{p}_{m_2}\| \overset{(B.11)}{\leq} 4\Upsilon_{upper}^{j} \cdot \frac{\Upsilon_{upper}^{t+1}}{\sqrt{\gamma_C}}\|\boldsymbol{p}_{m_2}\|.\end{aligned}$$

The above inequality also holds for $j=0$. Thus,

$$\|\boldsymbol{p}_{l:l+t_l-1}\|^2 = \sum_{j=0}^{t_l-1}\|\boldsymbol{p}_{l+j}\|^2 \le 16\sum_{j=0}^{t_l-1}\Upsilon_{upper}^{2j}\cdot\frac{\Upsilon_{upper}^{2t+2}}{\gamma_C}\|\boldsymbol{p}_{m_2}\|^2$$
$$= \frac{16(\Upsilon_{upper}^{2t}-1)\Upsilon_{upper}^{2t+2}}{\gamma_C(\Upsilon_{upper}^2-1)}\|\boldsymbol{p}_{m_2}\|^2 \le \frac{31\Upsilon_{upper}^{4t}}{\gamma_C}\|\boldsymbol{p}_{m_2}\|^2, \quad \text{(B.12)}$$

where the last inequality uses $\Upsilon_{upper}^2/(\Upsilon_{upper}^2-1)\le 31/16$ (as $\Upsilon_{upper}\ge 2$). Combining (B.12) with (B.9) and noting that $2t+2\le 4t$, we get

$$\|(\boldsymbol{p}_{l:l+t_l-1};\boldsymbol{q}_{l:l+t_l-1})\|^2 \le \frac{32\Upsilon_{upper}^{4t}}{\gamma_C}\|\boldsymbol{p}_{m_2}\|^2.$$

Finally, by a similar derivation as (B.5), we know $\tilde{\boldsymbol{w}}_i^T\hat{H}^{(i)}\tilde{\boldsymbol{w}}_i \ge \gamma_{RH}\|\tilde{\boldsymbol{w}}_i\|^2$ provided $\mu \ge 32\Upsilon_{upper}^{4t+1}/\gamma_C$. This condition implies (B.6), so we complete the proof.

B.3. Proof of Corollary 4.1 We note that all three conditions are independent of $\boldsymbol{d}_i$, so the statement holds for any $\boldsymbol{d}_i$. By Lemma 4.2, the reduced Hessian of $\mathcal{LP}_\mu^i(\boldsymbol{d}_i)$ is lower bounded by $\gamma_{RH}I$ for any $i\in[M-1]$. Thus, (i) holds. The controllability in Assumption 4.2 holds naturally for the subproblems with the same constants (γ_C,t), as the dynamics of the subproblems are a subset of the full problem. Thus, (ii) holds. For the upper boundedness condition, we note that the last square matrix of the objective of (3.9a) is bounded by $\Upsilon_{upper}+\mu$, and other square matrices are the same as the full problem (3.8). Thus, (iii) holds. This completes the proof.

B.4. Proof of Theorem 4.1 Our proof relies on the primal-dual sensitivity results of NLDPs [34, 38]. We omit the subproblem index i in the proof. We define

$$\tilde{g}_k(\boldsymbol{p}_k,\boldsymbol{q}_k) = \frac{1}{2}\begin{pmatrix}\boldsymbol{p}_k\\ \boldsymbol{q}_k\end{pmatrix}^T\begin{pmatrix}\hat{Q}_k & \hat{S}_k^T\\ \hat{S}_k & \hat{R}_k\end{pmatrix}\begin{pmatrix}\boldsymbol{p}_k\\ \boldsymbol{q}_k\end{pmatrix} + \begin{pmatrix}\nabla_{\boldsymbol{x}_k}\mathcal{L}\\ \nabla_{\boldsymbol{u}_k}\mathcal{L}\end{pmatrix}^T\begin{pmatrix}\boldsymbol{p}_k\\ \boldsymbol{q}_k\end{pmatrix}, \quad \forall k\in[m_1,m_2),$$
$$\tilde{g}_{m_2}(\boldsymbol{p}_{m_2};\boldsymbol{d}_{2:4}) = \frac{1}{2}\begin{pmatrix}\boldsymbol{p}_{m_2}\\ \boldsymbol{d}_3\end{pmatrix}^T\begin{pmatrix}\hat{Q}_{m_2} & \hat{S}_{m_2}^T\\ \hat{S}_{m_2} & \hat{R}_{m_2}\end{pmatrix}\begin{pmatrix}\boldsymbol{p}_{m_2}\\ \boldsymbol{d}_3\end{pmatrix} + \nabla_{\boldsymbol{x}_{m_2}}^T\mathcal{L}\boldsymbol{p}_{m_2} - \boldsymbol{d}_4^T A_{m_2}\boldsymbol{p}_{m_2} + \frac{\mu}{2}\|\boldsymbol{p}_{m_2}-\boldsymbol{d}_2\|^2,$$
$$\tilde{f}_k(\boldsymbol{p}_k,\boldsymbol{q}_k) = A_k\boldsymbol{p}_k + B_k\boldsymbol{q}_k - \nabla_{\boldsymbol{\lambda}_{k+1}}\mathcal{L}, \quad k\in[m_1,m_2).$$

Then, (3.9) is rewritten as $\min\sum_{k=m_1}^{m_2-1} g_k(\boldsymbol{p}_k,\boldsymbol{q}_k)+\tilde{g}_{m_2}(\boldsymbol{p}_{m_2};\boldsymbol{d}_{2:4})$, s.t. $\boldsymbol{p}_{m_1}=\boldsymbol{d}_1$, $\boldsymbol{p}_{k+1}=\tilde{f}_k(\boldsymbol{p}_k,\boldsymbol{q}_k)$, $\forall k\in[m_1,m_2)$. By Lemma 4.2, this problem has a unique solution $(\tilde{\boldsymbol{w}}^\star(\boldsymbol{d}),\tilde{\boldsymbol{\zeta}}^\star(\boldsymbol{d}))$ for any $\boldsymbol{d}$. Let us define a parameterized perturbation path from $\boldsymbol{d}$ to $\boldsymbol{d}'$:

$$\boldsymbol{d}^{(1)}(\omega) = (\boldsymbol{d}_1;\boldsymbol{d}_{2:4}) + \omega\left(\frac{\boldsymbol{d}_1'-\boldsymbol{d}_1}{\|\boldsymbol{d}_1'-\boldsymbol{d}_1\|};\boldsymbol{0}\right), \qquad \forall 0\le\omega\le\|\boldsymbol{d}_1'-\boldsymbol{d}_1\|,$$
$$\boldsymbol{d}^{(2)}(\omega) = (\boldsymbol{d}_1';\boldsymbol{d}_{2:4}) + \omega\left(\boldsymbol{0};\frac{\boldsymbol{d}_{2:4}'-\boldsymbol{d}_{2:4}}{\|\boldsymbol{d}_{2:4}'-\boldsymbol{d}_{2:4}\|}\right), \quad \forall 0\le\omega\le\|\boldsymbol{d}_{2:4}'-\boldsymbol{d}_{2:4}\|,$$

where we essentially first perturb $\boldsymbol{d}_1$ and then perturb $\boldsymbol{d}_{2:4}$. At $\boldsymbol{d}^{(j)}(\omega)$ for $j=1,2$, we define the directional derivatives of the solution trajectories as (similar for $D\boldsymbol{q}_k^\star, D\boldsymbol{\zeta}_k^\star$)

$$D\boldsymbol{p}_k^\star(\boldsymbol{d}^{(j)}(\omega)) = \lim_{\epsilon\searrow 0}\frac{\boldsymbol{p}_k^\star(\boldsymbol{d}^{(j)}(\omega+\epsilon))-\boldsymbol{p}_k^\star(\boldsymbol{d}^{(j)}(\omega))}{\epsilon}, \qquad \forall k\in[m_1,m_2].$$

The existence of the directional derivatives is ensured by Lemma 4.2 and [34, Theorem 2.3]. By Corollary 4.1 and the fact that (by Assumption 4.3)

$$\|\nabla_{\boldsymbol{p}_{m_2}\boldsymbol{d}_{2:4}}\tilde{g}_{m_2}\| = \|(\mu I \;\; S_{m_2}^T \;\; -A_{m_2}^T)\| \le \mu + 2\Upsilon_{upper},$$

we know that [34, Assumption 4.2] is satisfied. Thus, by [34, Theorem 5.7],

$$\|D\boldsymbol{p}_k^\star(\boldsymbol{d}^{(1)}(\omega))\| \le C'\rho^{k-m_1}, \quad \|D\boldsymbol{p}_k^\star(\boldsymbol{d}^{(2)}(\omega))\| \le C'\rho^{m_2-k}, \quad \forall k \in [m_1, m_2], \tag{B.13}$$

for constants $C' > 0$ and $\rho \in (0,1)$ depending on $\mu, \Upsilon_{upper}, \gamma_{RH}, \gamma_C$ only. By [34, Theorem 5.7] and [38, Theorem 5], we know (B.13) holds for $D\boldsymbol{q}_k^\star(\boldsymbol{d}^{(j)}(\omega))$ and $D\boldsymbol{\zeta}_k^\star(\boldsymbol{d}^{(j)}(\omega))$ as well. Furthermore,

$$\begin{aligned}
\|\boldsymbol{p}_k^\star(\boldsymbol{d}) - \boldsymbol{p}_k^\star(\boldsymbol{d}')\| &= \|\boldsymbol{p}_k^\star(\boldsymbol{d}^{(1)}(0)) - \boldsymbol{p}_k^\star(\boldsymbol{d}^{(2)}(\|\boldsymbol{d}'_{2:4} - \boldsymbol{d}_{2:4}\|))\| \\
&\le \|\boldsymbol{p}_k^\star(\boldsymbol{d}^{(1)}(0)) - \boldsymbol{p}_k^\star(\boldsymbol{d}^{(1)}(\|\boldsymbol{d}'_1 - \boldsymbol{d}_1\|))\| + \|\boldsymbol{p}_k^\star(\boldsymbol{d}^{(2)}(0)) - \boldsymbol{p}_k^\star(\boldsymbol{d}^{(2)}(\|\boldsymbol{d}'_{2:4} - \boldsymbol{d}_{2:4}\|))\| \\
&= \left\| \int_0^{\|\boldsymbol{d}'_1 - \boldsymbol{d}_1\|} D\boldsymbol{p}_k^\star(\boldsymbol{d}^{(1)}(\omega)) d\omega \right\| + \left\| \int_0^{\|\boldsymbol{d}'_{2:4} - \boldsymbol{d}_{2:4}\|} D\boldsymbol{p}_k^\star(\boldsymbol{d}^{(2)}(\omega)) d\omega \right\| \\
&\overset{(B.13)}{\le} C' \left(\rho^{k-m_1} \|\boldsymbol{d}'_1 - \boldsymbol{d}_1\| + \rho^{m_2-k} \|\boldsymbol{d}'_{2:4} - \boldsymbol{d}_{2:4}\| \right).
\end{aligned}$$

The above inequality also holds for $\|\boldsymbol{q}_k^\star(\boldsymbol{d}) - \boldsymbol{q}_k^\star(\boldsymbol{d}')\|$, $\|\boldsymbol{\zeta}_k^\star(\boldsymbol{d}) - \boldsymbol{\zeta}_k^\star(\boldsymbol{d}')\|$. Since $\|\boldsymbol{w}_k^\star(\boldsymbol{d}) - \boldsymbol{w}_k^\star(\boldsymbol{d}')\| = \sqrt{\|\boldsymbol{p}_k^\star(\boldsymbol{d}) - \boldsymbol{p}_k^\star(\boldsymbol{d}')\|^2 + \|\boldsymbol{q}_k^\star(\boldsymbol{d}) - \boldsymbol{q}_k^\star(\boldsymbol{d}')\|^2}$, the proof is complete by redefining $C' \leftarrow \sqrt{2}C'$.

B.5. Proof of Theorem 4.2 Since $(\tilde{\Delta}\boldsymbol{z}, \tilde{\Delta}\boldsymbol{\lambda}) = \mathcal{C}(\{(\tilde{\boldsymbol{w}}_i^\star(\boldsymbol{d}_i), \tilde{\boldsymbol{\zeta}}_i^\star(\boldsymbol{d}_i))\}_i)$ with $\boldsymbol{d}_i = (\boldsymbol{0}; \boldsymbol{0}; \boldsymbol{0}; \boldsymbol{0})$ ($\boldsymbol{d}_{M-1} = \boldsymbol{0}$), by Definition 2.1 we know that

$$\begin{aligned}
&\tilde{\Delta}\boldsymbol{z}_k = \tilde{\boldsymbol{w}}_{i,k}^\star(\boldsymbol{d}_i), \quad \text{if } k \in [n_i, n_{i+1}) \text{ for some } i \in [M-1], \qquad \tilde{\Delta}\boldsymbol{z}_N = \tilde{\boldsymbol{w}}_{M-1,N}^\star(\boldsymbol{d}_{M-1}), \\
&\tilde{\Delta}\boldsymbol{\lambda}_k = \tilde{\boldsymbol{\zeta}}_{i,k}^\star(\boldsymbol{d}_i), \quad \text{if } k \in [n_i, n_{i+1}) \text{ for some } i \in [M-1], \qquad \tilde{\Delta}\boldsymbol{\lambda}_N = \tilde{\boldsymbol{\zeta}}_{M-1,N}^\star(\boldsymbol{d}_{M-1}).
\end{aligned} \tag{B.14}$$

Applying Theorem 2.1(i) on Problem (3.8), we know that if

$$\boldsymbol{d}_i^\star = (\Delta\boldsymbol{x}_{m_1}; \Delta\boldsymbol{x}_{m_2}; \Delta\boldsymbol{u}_{m_2}; \Delta\boldsymbol{\lambda}_{m_2+1}), \tag{B.15}$$

then

$$\begin{aligned}
&\Delta\boldsymbol{z}_k = \tilde{\boldsymbol{w}}_{i,k}^\star(\boldsymbol{d}_i^\star), \quad \text{if } k \in [n_i, n_{i+1}) \text{ for some } i \in [M-1], \qquad \Delta\boldsymbol{z}_N = \tilde{\boldsymbol{w}}_{M-1,N}^\star(\boldsymbol{d}_{M-1}^\star), \\
&\Delta\boldsymbol{\lambda}_k = \tilde{\boldsymbol{\zeta}}_{i,k}^\star(\boldsymbol{d}_i^\star), \quad \text{if } k \in [n_i, n_{i+1}) \text{ for some } i \in [M-1], \qquad \Delta\boldsymbol{\lambda}_N = \tilde{\boldsymbol{\zeta}}_{M-1,N}^\star(\boldsymbol{d}_{M-1}^\star).
\end{aligned} \tag{B.16}$$

Moreover, we claim $\boldsymbol{d}_{0,1}^\star = \Delta\boldsymbol{x}_0 = \boldsymbol{0}$ for any iteration τ. In fact, from the input to Algorithm 2, $\boldsymbol{x}_0^0 = \bar{\boldsymbol{x}}_0$. By (3.8c), $\Delta\boldsymbol{x}_0^0 = -(\boldsymbol{x}_0^0 - \bar{\boldsymbol{x}}_0) = \boldsymbol{0}$. Furthermore, if $\boldsymbol{x}_0^{\tau-1} = \bar{\boldsymbol{x}}_0$ for $\tau \ge 1$, we use the fact that $\tilde{\Delta}\boldsymbol{x}_0^{\tau-1} = \boldsymbol{d}_{0,1}^{\tau-1} = \boldsymbol{0}$ (the first equality is due to (3.9c); the second equality is due to the specification of the boundary variable), and obtain $\boldsymbol{x}_0^\tau = \boldsymbol{x}_0^{\tau-1} + \alpha_{\tau-1}\tilde{\Delta}\boldsymbol{x}^{\tau-1} = \boldsymbol{x}_0^{\tau-1} = \bar{\boldsymbol{x}}_0$. Thus, we have $\Delta\boldsymbol{x}_0^\tau \overset{(3.8c)}{=} -(\boldsymbol{x}_0^\tau - \bar{\boldsymbol{x}}_0) = \boldsymbol{0}$. Comparing (B.14) and (B.16) for each stage and applying Theorem 4.1, we have

$$\begin{aligned}
&\max\{\|\tilde{\Delta}\boldsymbol{z}_k - \Delta\boldsymbol{z}_k\|^2, \|\tilde{\Delta}\boldsymbol{\lambda}_k - \Delta\boldsymbol{\lambda}_k\|^2\} \\
&\quad \le 2(C')^2(\rho^{2(k-m_1)}\|\boldsymbol{d}_{i,1}^\star\|^2 + \rho^{2(m_2-k)}\|\boldsymbol{d}_{i,2:4}^\star\|^2), \qquad \forall k \in [n_i, n_{i+1}),\ i \in [M-2], \qquad \text{(B.17a)} \\
&\max\{\|\tilde{\Delta}\boldsymbol{z}_k - \Delta\boldsymbol{z}_k\|^2, \|\tilde{\Delta}\boldsymbol{\lambda}_k - \Delta\boldsymbol{\lambda}_k\|^2\} \le (C')^2\rho^{2(k-m_1)}\|\boldsymbol{d}_{i,1}^\star\|^2, \qquad \forall k \in [n_{M-1}, n_M], \qquad \text{(B.17b)}
\end{aligned}$$

where (B.17a) holds for $i = 0$ since $\boldsymbol{d}_{0,1}^\star = \boldsymbol{0}$ as we just claimed. Thus, we get

$$\begin{aligned}
\|\tilde{\Delta}\boldsymbol{z} - \Delta\boldsymbol{z}\|^2 &= \sum_{i=0}^{M-2} \sum_{k=n_i}^{n_{i+1}-1} \|\tilde{\Delta}\boldsymbol{z}_k - \Delta\boldsymbol{z}_k\|^2 + \sum_{k=n_{M-1}}^{n_M} \|\tilde{\Delta}\boldsymbol{z}_k - \Delta\boldsymbol{z}_k\|^2 \\
&\overset{(B.17)}{\le} 2(C')^2 \sum_{i=0}^{M-2} \left(\sum_{j=0}^{n_{i+1}-n_i-1} \rho^{2(b+j)}\|\boldsymbol{d}_{i,1}^\star\|^2 + \sum_{j=1}^{n_{i+1}-n_i} \rho^{2(b+j)}\|\boldsymbol{d}_{i,2:4}^\star\|^2 \right) + (C')^2 \sum_{j=0}^{n_M - n_{M-1}} \rho^{2(b+j)}\|\boldsymbol{d}_{M-1}^\star\|^2 \\
&\le 2(C')^2\rho^{2b} \sum_{i=0}^{M-2} \sum_{j=0}^{\infty} \rho^{2j}\|\boldsymbol{d}_i^\star\|^2 + (C')^2\rho^{2b} \sum_{j=0}^{\infty} \rho^{2j}\|\boldsymbol{d}_{M-1}^\star\|^2 \\
&\le \frac{2(C')^2\rho^{2b}}{1-\rho^2} \sum_{i=0}^{M-1} \|\boldsymbol{d}_i^\star\|^2 \overset{(B.15)}{\le} \frac{2(C')^2\rho^{2b}}{1-\rho^2} \|(\Delta\boldsymbol{z}, \Delta\boldsymbol{\lambda})\|^2.
\end{aligned}$$

The above derivation also holds for $\|\tilde{\Delta}\boldsymbol{\lambda} - \Delta\boldsymbol{\lambda}\|^2$. Thus,

$$\|(\tilde{\Delta}\boldsymbol{z} - \Delta\boldsymbol{z}, \tilde{\Delta}\boldsymbol{\lambda} - \Delta\boldsymbol{\lambda})\|^2 \leq \frac{4(C')^2\rho^{2b}}{1-\rho^2}\|(\Delta\boldsymbol{z}, \Delta\boldsymbol{\lambda})\|^2.$$

Letting $C = 2C'/\sqrt{1-\rho^2}$, we complete the proof.

Appendix C: Proofs of results in section 5

C.1. Proof of Lemma 5.1 Recall from (3.7) that $H(\boldsymbol{z}, \boldsymbol{\lambda}) = \operatorname{diag}(H_0, \ldots, H_N)$. Thus,

$$\|H(\boldsymbol{z}, \boldsymbol{\lambda})\| \leq \max_{k\in[N]} \|H_k(\boldsymbol{z}_k, \boldsymbol{\lambda}_{k+1})\| \leq \Upsilon_{upper}$$

for any $(\boldsymbol{z}, \boldsymbol{\lambda}) \in \mathcal{Z} \times \Lambda$, where the last inequality is due to Assumption 5.1. Furthermore, using the expression of GG^T in (B.1), we immediately have

$$\|G(\boldsymbol{z})\| = \sqrt{\|GG^T\|} \leq \sqrt{1 + \|A_k\|^2 + \|B_k^2\| + 2\|A_k\|} \overset{(5.2)}{\leq} 1 + 2\Upsilon_{upper}.$$

Thus, we can let $\Upsilon_{HG} = 1 + 2\Upsilon_{upper}$ and the first part of the statement holds. Moreover, if $\|\hat{H}_k\| \leq \Upsilon_{upper}$ by Assumption 4.3, then $\|\hat{H}\| = \|\operatorname{diag}(\hat{H}_0, \ldots, \hat{H}_N)\| \leq \Upsilon_{upper}$. Let Z be defined in Assumption 4.1. Then, noting that $G^T(GG^T)^{-1}G + ZZ^T = I$, we can verify that

$$\mathcal{B} := \begin{pmatrix} \hat{H} & G^T \\ G & \mathbf{0} \end{pmatrix}^{-1} = \begin{pmatrix} \mathcal{B}_1 & \mathcal{B}_2^T \\ \mathcal{B}_2 & \mathcal{B}_3 \end{pmatrix},$$

where

$$\begin{aligned} \mathcal{B}_1 &= Z(Z^T\hat{H}Z)^{-1}Z^T, \qquad \mathcal{B}_2 = (GG^T)^{-1}G(I - \hat{H}Z(Z^T\hat{H}Z)^{-1}Z^T), \\ \mathcal{B}_3 &= (GG^T)^{-1}G(\hat{H}Z(Z^T\hat{H}Z)^{-1}Z^T\hat{H} - \hat{H})G^T(GG^T)^{-1}. \end{aligned}$$

Then, by Assumptions 4.1-4.3 and Lemma 4.1, we have

$$\begin{aligned} \|\mathcal{B}_1\| &\leq \frac{1}{\gamma_{RH}}, \quad \|\mathcal{B}_2\| \leq \|(GG^T)^{-1}G\|(1 + \frac{\Upsilon_{upper}}{\gamma_{RH}}) \leq \frac{1}{\sqrt{\gamma_G}}(1 + \frac{\Upsilon_{upper}}{\gamma_{RH}}), \\ \|\mathcal{B}_3\| &\leq \frac{1}{\gamma_G}(\Upsilon_{upper} + \frac{\Upsilon_{upper}^2}{\gamma_{RH}}). \end{aligned}$$

Noting that $\|\mathcal{B}\| \leq \|\mathcal{B}_1\| + 2\|\mathcal{B}_2\| + \|\mathcal{B}_3\|$, we complete the proof.

C.2. Proof of Theorem 5.1 We suppress the iteration index τ. We know

$$\begin{pmatrix} \nabla_{\boldsymbol{z}}\mathcal{L}_\eta \\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta \end{pmatrix}^T \begin{pmatrix} \tilde{\Delta}\boldsymbol{z} \\ \tilde{\Delta}\boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \nabla_{\boldsymbol{z}}\mathcal{L}_\eta \\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta \end{pmatrix}^T \begin{pmatrix} \Delta\boldsymbol{z} \\ \Delta\boldsymbol{\lambda} \end{pmatrix} + \begin{pmatrix} \nabla_{\boldsymbol{z}}\mathcal{L}_\eta \\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta \end{pmatrix}^T \begin{pmatrix} \tilde{\Delta}\boldsymbol{z} - \Delta\boldsymbol{z} \\ \tilde{\Delta}\boldsymbol{\lambda} - \Delta\boldsymbol{\lambda} \end{pmatrix}. \tag{C.1}$$

For the first term in (C.1),

$$\begin{aligned}
\mathcal{I}_1 &\coloneqq \begin{pmatrix}\nabla_{\boldsymbol{z}}\mathcal{L}_\eta\\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta\end{pmatrix}^T\begin{pmatrix}\Delta\boldsymbol{z}\\ \Delta\boldsymbol{\lambda}\end{pmatrix}\\
&\overset{(3.6)}{=}\begin{pmatrix}\Delta\boldsymbol{z}\\ \Delta\boldsymbol{\lambda}\end{pmatrix}^T\begin{pmatrix}I+\eta_2 H & \eta_1 G^T\\ \eta_2 G & I\end{pmatrix}\begin{pmatrix}\nabla_{\boldsymbol{z}}\mathcal{L}\\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}\end{pmatrix}\\
&\overset{(3.2)}{=}-\begin{pmatrix}\Delta\boldsymbol{z}\\ \Delta\boldsymbol{\lambda}\end{pmatrix}^T\begin{pmatrix}I+\eta_2 H & \eta_1 G^T\\ \eta_2 G & I\end{pmatrix}\begin{pmatrix}\hat{H} & G^T\\ G & \mathbf{0}\end{pmatrix}\begin{pmatrix}\Delta\boldsymbol{z}\\ \Delta\boldsymbol{\lambda}\end{pmatrix}\\
&=-\begin{pmatrix}\Delta\boldsymbol{z}\\ \Delta\boldsymbol{\lambda}\end{pmatrix}^T\begin{pmatrix}(I+\eta_2 H)\hat{H}+\eta_1 G^T G & (I+\eta_2 H)G^T\\ G(I+\eta_2\hat{H}) & \eta_2 GG^T\end{pmatrix}\begin{pmatrix}\Delta\boldsymbol{z}\\ \Delta\boldsymbol{\lambda}\end{pmatrix}\\
&\overset{(3.2)}{=}-(\Delta\boldsymbol{z})^T\left\{(I+\eta_2 H)\hat{H}+\frac{\eta_1}{2}G^T G\right\}\Delta\boldsymbol{z}-\frac{\eta_1}{2}\|\nabla_{\boldsymbol{\lambda}}\mathcal{L}\|^2-\eta_2\|\hat{H}\Delta\boldsymbol{z}+\nabla_{\boldsymbol{z}}\mathcal{L}\|^2\\
&\quad-(\Delta\boldsymbol{\lambda})^T G\{2I+\eta_2(\hat{H}+H)\}\Delta\boldsymbol{z}\\
&=-\frac{\eta_1}{2}\|\nabla_{\boldsymbol{\lambda}}\mathcal{L}\|^2-\frac{\eta_2}{2}\|\nabla_{\boldsymbol{z}}\mathcal{L}\|^2-(\Delta\boldsymbol{z})^T\left\{(I+\eta_2 H)\hat{H}+\frac{\eta_1}{2}G^T G\right\}\Delta\boldsymbol{z}\\
&\quad-(\Delta\boldsymbol{\lambda})^T G\{2I+\eta_2(\hat{H}+H)\}\Delta\boldsymbol{z}+\left(\frac{\eta_2}{2}\|\nabla_{\boldsymbol{z}}\mathcal{L}\|^2-\eta_2\|\hat{H}\Delta\boldsymbol{z}+\nabla_{\boldsymbol{z}}\mathcal{L}\|^2\right).
\end{aligned}\tag{C.2}$$

For the last term in the above equation,

$$\begin{aligned}
&\frac{\eta_2}{2}\|\nabla_{\boldsymbol{z}}\mathcal{L}\|^2-\eta_2\|\hat{H}\Delta\boldsymbol{z}+\nabla_{\boldsymbol{z}}\mathcal{L}\|^2\\
&\quad=-\eta_2\|\hat{H}\Delta\boldsymbol{z}\|^2-2\eta_2(\Delta\boldsymbol{z})^T\hat{H}\nabla_{\boldsymbol{z}}\mathcal{L}-\frac{\eta_2}{2}\|\nabla_{\boldsymbol{z}}\mathcal{L}\|^2\\
&\quad\overset{(3.2)}{=}-\eta_2\|\hat{H}\Delta\boldsymbol{z}\|^2+2\eta_2(\Delta\boldsymbol{z})^T\hat{H}(\hat{H}\Delta\boldsymbol{z}+G^T\Delta\boldsymbol{\lambda})-\frac{\eta_2}{2}\|\hat{H}\Delta\boldsymbol{z}+G^T\Delta\boldsymbol{\lambda}\|^2\\
&\quad=\eta_2\|\hat{H}\Delta\boldsymbol{z}\|^2+\eta_2(\Delta\boldsymbol{z})^T\hat{H}G^T\Delta\boldsymbol{\lambda}-\frac{\eta_2}{2}\|\hat{H}\Delta\boldsymbol{z}\|^2-\frac{\eta_2}{2}\|G^T\Delta\boldsymbol{\lambda}\|^2\\
&\quad\le\eta_2\|\hat{H}\Delta\boldsymbol{z}\|^2+\eta_2(\Delta\boldsymbol{z})^T\hat{H}G^T\Delta\boldsymbol{\lambda}-\frac{\eta_2}{2}\|G^T\Delta\boldsymbol{\lambda}\|^2.
\end{aligned}\tag{C.3}$$

Combining the above two displays and supposing $\eta_1\ge\eta_2$ at the moment,

$$\begin{aligned}
\mathcal{I}_1&\overset{(C.2)}{\le}-\frac{\eta_2}{2}\|\nabla\mathcal{L}\|^2-(\Delta\boldsymbol{z})^T\left\{(I+\eta_2 H)\hat{H}+\frac{\eta_1}{2}G^T G\right\}\Delta\boldsymbol{z}-(\Delta\boldsymbol{\lambda})^T G\{2I+\eta_2(\hat{H}+H)\}\Delta\boldsymbol{z}\\
&\quad+\left(\frac{\eta_2}{2}\|\nabla_{\boldsymbol{z}}\mathcal{L}\|^2-\eta_2\|\hat{H}\Delta\boldsymbol{z}+\nabla_{\boldsymbol{z}}\mathcal{L}\|^2\right)\\
&\overset{(C.3)}{\le}-\frac{\eta_2}{2}\|\nabla\mathcal{L}\|^2-(\Delta\boldsymbol{z})^T\left\{(I+\eta_2 H)\hat{H}+\frac{\eta_1}{2}G^T G\right\}\Delta\boldsymbol{z}-(\Delta\boldsymbol{\lambda})^T G\{2I+\eta_2(\hat{H}+H)\}\Delta\boldsymbol{z}\\
&\quad+\eta_2\|\hat{H}\Delta\boldsymbol{z}\|^2+\eta_2(\Delta\boldsymbol{z})^T\hat{H}G^T\Delta\boldsymbol{\lambda}-\frac{\eta_2}{2}\|G^T\Delta\boldsymbol{\lambda}\|^2\\
&=-\frac{\eta_2}{2}\|\nabla\mathcal{L}\|^2-(\Delta\boldsymbol{z})^T\left\{(I+\eta_2(H-\hat{H}))\hat{H}+\frac{\eta_1}{2}G^T G\right\}\Delta\boldsymbol{z}-(\Delta\boldsymbol{\lambda})^T G(2I+\eta_2 H)\Delta\boldsymbol{z}-\frac{\eta_2}{2}\|G^T\Delta\boldsymbol{\lambda}\|^2\\
&\le-\frac{\eta_2}{2}\|\nabla\mathcal{L}\|^2+2\eta_2\Upsilon^2\|\Delta\boldsymbol{z}\|^2+2\|\Delta\boldsymbol{\lambda}\|\|G\Delta\boldsymbol{z}\|+\eta_2\Upsilon\|G^T\Delta\boldsymbol{\lambda}\|\|\Delta\boldsymbol{z}\|-\frac{\eta_2}{2}\|G^T\Delta\boldsymbol{\lambda}\|^2\\
&\quad-(\Delta\boldsymbol{z})^T\left\{\hat{H}+\frac{\eta_1}{2}G^T G\right\}\Delta\boldsymbol{z}\\
&\le-\frac{\eta_2}{2}\|\nabla\mathcal{L}\|^2+3\eta_2\Upsilon^2\|\Delta\boldsymbol{z}\|^2+2\|\Delta\boldsymbol{\lambda}\|\|G\Delta\boldsymbol{z}\|-\frac{\eta_2}{4}\|G^T\Delta\boldsymbol{\lambda}\|^2-(\Delta\boldsymbol{z})^T\left\{\hat{H}+\frac{\eta_1}{2}G^T G\right\}\Delta\boldsymbol{z}\\
&\overset{\text{Lemma 4.1}}{\le}-\frac{\eta_2}{2}\|\nabla\mathcal{L}\|^2+3\eta_2\Upsilon^2\|\Delta\boldsymbol{z}\|^2+2\|\Delta\boldsymbol{\lambda}\|\|G\Delta\boldsymbol{z}\|-\frac{\eta_2\gamma_G}{4}\|\Delta\boldsymbol{\lambda}\|^2-(\Delta\boldsymbol{z})^T\left\{\hat{H}+\frac{\eta_1}{2}G^T G\right\}\Delta\boldsymbol{z}\\
&\le-\frac{\eta_2}{2}\|\nabla\mathcal{L}\|^2-\frac{\eta_2\gamma_G}{8}\|\Delta\boldsymbol{\lambda}\|^2+3\eta_2\Upsilon^2\|\Delta\boldsymbol{z}\|^2-(\Delta\boldsymbol{z})^T\left\{\hat{H}+\left(\frac{\eta_1}{2}-\frac{8}{\eta_2\gamma_G}\right)G^T G\right\}\Delta\boldsymbol{z},
\end{aligned}\tag{C.4}$$

where the fourth inequality uses Assumption 4.3 and (5.3) so that $\max\{\|\hat{H}\|, \|H\|\} \le \Upsilon$; the fifth and seventh inequalities use Young's inequalities:

$$\eta_2\Upsilon\|G^T\Delta\boldsymbol{\lambda}\|\|\Delta\boldsymbol{z}\| \le \frac{\eta_2}{4}\|G^T\Delta\boldsymbol{\lambda}\|^2 + \eta_2\Upsilon^2\|\Delta\boldsymbol{z}\|^2,$$
$$2\|\Delta\boldsymbol{\lambda}\|\|G\Delta\boldsymbol{z}\| \le \frac{\eta_2\gamma_G}{8}\|\Delta\boldsymbol{\lambda}\|^2 + \frac{8}{\eta_2\gamma_G}\|G\Delta\boldsymbol{z}\|^2.$$

To simplify the last two terms, we suppose $\eta_1 \ge 16/(\eta_2\gamma_G)$, and decompose $\Delta\boldsymbol{z} = \Delta\boldsymbol{u} + \Delta\boldsymbol{v}$, where $\Delta\boldsymbol{u} \in \text{span}(G^T)$ so that $\Delta\boldsymbol{u} = G^T\bar{\Delta}\boldsymbol{u}$ for some $\bar{\Delta}\boldsymbol{u}$, and $\Delta\boldsymbol{v} \in \text{kernel}(G)$ so that $G\Delta\boldsymbol{v} = \boldsymbol{0}$. Then, we have $\|\Delta\boldsymbol{z}\|^2 = \|\Delta\boldsymbol{u}\|^2 + \|\Delta\boldsymbol{v}\|^2$ and

$$\begin{aligned}
&3\eta_2\Upsilon^2\|\Delta\boldsymbol{z}\|^2 - (\Delta\boldsymbol{z})^T\left\{\hat{H} + \left(\frac{\eta_1}{2} - \frac{8}{\eta_2\gamma_G}\right)G^TG\right\}\Delta\boldsymbol{z} \\
&\quad= 3\eta_2\Upsilon^2\|\Delta\boldsymbol{z}\|^2 - (\Delta\boldsymbol{v})^T\hat{H}\Delta\boldsymbol{v} - 2(\Delta\boldsymbol{v})^T\hat{H}\Delta\boldsymbol{u} - (\Delta\boldsymbol{u})^T\hat{H}\Delta\boldsymbol{u} - \left(\frac{\eta_1}{2} - \frac{8}{\eta_2\gamma_G}\right)\|GG^T\bar{\Delta}\boldsymbol{u}\|^2 \\
&\quad\le (3\eta_2\Upsilon^2 - \gamma_{RH})\|\Delta\boldsymbol{z}\|^2 + 2\Upsilon\|\Delta\boldsymbol{v}\|\|\Delta\boldsymbol{u}\| + (\gamma_{RH} + \Upsilon)\|\Delta\boldsymbol{u}\|^2 - \left(\frac{\eta_1}{2} - \frac{8}{\eta_2\gamma_G}\right)\gamma_G\|G^T\bar{\Delta}\boldsymbol{u}\|^2 \\
&\quad\le \left(3\eta_2\Upsilon^2 - \frac{\gamma_{RH}}{2}\right)\|\Delta\boldsymbol{z}\|^2 + \left(\gamma_{RH} + \Upsilon + \frac{2\Upsilon^2}{\gamma_{RH}} + \frac{8}{\eta_2} - \frac{\eta_1\gamma_G}{2}\right)\|\Delta\boldsymbol{u}\|^2, \qquad \text{(C.5)}
\end{aligned}$$

where the second inequality uses Assumptions 4.1, 4.3, and Lemma 4.1, and the third inequality uses Young's inequality

$$2\Upsilon\|\Delta\boldsymbol{v}\|\|\Delta\boldsymbol{u}\| \le \frac{\gamma_{RH}}{2}\|\Delta\boldsymbol{v}\|^2 + \frac{2\Upsilon^2}{\gamma_{RH}}\|\Delta\boldsymbol{u}\|^2 \le \frac{\gamma_{RH}}{2}\|\Delta\boldsymbol{z}\|^2 + \frac{2\Upsilon^2}{\gamma_{RH}}\|\Delta\boldsymbol{u}\|^2.$$

To make (C.5) negative, we let

$$3\eta_2\Upsilon^2 \le \frac{\gamma_{RH}}{4} \iff \eta_2 \le \frac{\gamma_{RH}}{12\Upsilon^2}. \qquad \text{(C.6)}$$

Furthermore, without loss of generality, we suppose $\Upsilon/2 \ge 1 \ge \max\{\gamma_{RH}, \gamma_G\}$, and have

$$\gamma_{RH} + \Upsilon + \frac{2\Upsilon^2}{\gamma_{RH}} + \frac{8}{\eta_2} \le \frac{3\Upsilon}{2} + \frac{2\Upsilon^2}{\gamma_{RH}} + \frac{8}{\eta_2} \le \frac{3\Upsilon^2}{\gamma_{RH}} + \frac{8}{\eta_2} \overset{\text{(C.6)}}{\le} \frac{1}{4\eta_2} + \frac{8}{\eta_2} \le \frac{8.5}{\eta_2}.$$

Thus, we let

$$\eta_1 \ge \frac{17}{\eta_2\gamma_G}, \qquad \text{(C.7)}$$

which implies $\eta_1 \ge \eta_2$ and $\eta_1 \ge 16/(\eta_2\gamma_G)$ as required in (C.4) and (C.5). Thus, under (C.7) and (C.6), the inequality (C.5) leads to

$$3\eta_2\Upsilon^2\|\Delta\boldsymbol{z}\|^2 - (\Delta\boldsymbol{z})^T\left\{\hat{H} + \left(\frac{\eta_1}{2} - \frac{8}{\eta_2\gamma_G}\right)G^TG\right\}\Delta\boldsymbol{z} \le -\frac{\gamma_{RH}}{4}\|\Delta\boldsymbol{z}\|^2 \overset{\text{(C.6)}}{\le} -\frac{\eta_2\gamma_G}{8}\|\Delta\boldsymbol{z}\|^2.$$

Combining the above display with (C.4), we obtain

$$\mathcal{I}_1 = \begin{pmatrix}\nabla_{\boldsymbol{z}}\mathcal{L}_\eta \\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta\end{pmatrix}^T\begin{pmatrix}\Delta\boldsymbol{z} \\ \Delta\boldsymbol{\lambda}\end{pmatrix} \le -\frac{\eta_2}{2}\|\nabla\mathcal{L}\|^2 - \frac{\eta_2\gamma_G}{8}\left\|\begin{pmatrix}\Delta\boldsymbol{z} \\ \Delta\boldsymbol{\lambda}\end{pmatrix}\right\|^2. \qquad \text{(C.8)}$$

For the second term in (C.1),

$$\begin{aligned}
\mathcal{I}_2 &:= \begin{pmatrix}\nabla_{\boldsymbol{z}}\mathcal{L}_\eta \\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta\end{pmatrix}^T\begin{pmatrix}\tilde{\Delta}\boldsymbol{z} - \Delta\boldsymbol{z} \\ \tilde{\Delta}\boldsymbol{\lambda} - \Delta\boldsymbol{\lambda}\end{pmatrix} \overset{\substack{\text{(3.6)}\\\text{(3.2)}}}{=} -\begin{pmatrix}\tilde{\Delta}\boldsymbol{z} - \Delta\boldsymbol{z} \\ \tilde{\Delta}\boldsymbol{\lambda} - \Delta\boldsymbol{\lambda}\end{pmatrix}^T\begin{pmatrix}(I + \eta_2 H)\hat{H} + \eta_1 G^TG & (I + \eta_2 H)G^T \\ G(I + \eta_2\hat{H}) & \eta_2 GG^T\end{pmatrix}\begin{pmatrix}\Delta\boldsymbol{z} \\ \Delta\boldsymbol{\lambda}\end{pmatrix} \\
&\overset{\text{(5.1)}}{\le} \delta\|(\Delta\boldsymbol{z}, \Delta\boldsymbol{\lambda})\|^2\left\{\Upsilon + \eta_2\Upsilon^2 + (\eta_1 + \eta_2)\Upsilon^2 + 2(1 + \eta_2\Upsilon)\Upsilon\right\} = \delta\|(\Delta\boldsymbol{z}, \Delta\boldsymbol{\lambda})\|^2\left\{3\Upsilon + 4\eta_2\Upsilon^2 + \eta_1\Upsilon^2\right\},
\end{aligned}$$

where the third inequality also uses Assumption 4.3 and (5.3) so that $\max\{\|G\|, \|\hat{H}\|, \|H\|\} \le \Upsilon$. We use $\Upsilon/2 \ge 1 \ge \max\{\gamma_{RH}, \gamma_G\}$ and know that

$$3\Upsilon + 4\eta_2\Upsilon^2 + \eta_1\Upsilon^2 \overset{(C.6)}{\le} 3\Upsilon + \frac{\gamma_{RH}}{3} + \eta_1\Upsilon^2 \le \frac{9.5\Upsilon}{3} + \eta_1\Upsilon^2 \le 1.1\eta_1\Upsilon^2, \tag{C.9}$$

where the last inequality uses $\eta_1 \overset{(C.7)}{\ge} \frac{17}{\eta_2\gamma_G} > \frac{17\times 12\Upsilon^2}{\gamma_{RH}\gamma_G}$, so $9.5/3 \le 0.1\eta_1\Upsilon$. By the above two displays,

$$\mathcal{I}_2 = \begin{pmatrix}\nabla_{\boldsymbol{z}}\mathcal{L}_\eta \\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta\end{pmatrix}^T \begin{pmatrix}\tilde{\Delta}\boldsymbol{z} - \Delta\boldsymbol{z} \\ \tilde{\Delta}\boldsymbol{\lambda} - \Delta\boldsymbol{\lambda}\end{pmatrix} \le 1.1\eta_1\delta\Upsilon^2\|(\Delta\boldsymbol{z}, \Delta\boldsymbol{\lambda})\|^2. \tag{C.10}$$

Combining (C.10) with (C.8) and (C.1),

$$\begin{pmatrix}\nabla_{\boldsymbol{z}}\mathcal{L}_\eta \\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta\end{pmatrix}^T \begin{pmatrix}\tilde{\Delta}\boldsymbol{z} \\ \tilde{\Delta}\boldsymbol{\lambda}\end{pmatrix} = \mathcal{I}_1 + \mathcal{I}_2 \le -\frac{\eta_2}{2}\left\|\begin{pmatrix}\nabla_{\boldsymbol{z}}\mathcal{L} \\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}\end{pmatrix}\right\|^2 - \left(\frac{\eta_2\gamma_G}{8} - 1.1\eta_1\delta\Upsilon^2\right)\left\|\begin{pmatrix}\Delta\boldsymbol{z} \\ \Delta\boldsymbol{\lambda}\end{pmatrix}\right\|^2 \le -\frac{\eta_2}{2}\|\nabla\mathcal{L}\|^2, \tag{C.11}$$

where the last inequality holds if $\delta \le \frac{\eta_2\gamma_G}{9\eta_1\Upsilon^2}$. This completes the proof.

C.3. Proof of Theorem 5.2 It suffices to prove the first part of the statement. The second part holds immediately by Theorem 4.2 and specializes (5.1) with $\delta = C\rho^b$. By compactness of iterates and continuous differentiability of $\{g_k, f_k\}$ in Assumption 5.1, we know $\sup_{\mathcal{Z}\times\Lambda}\|\nabla^2\mathcal{L}_\eta(\boldsymbol{z}, \boldsymbol{\lambda})\| \le \Upsilon_\eta$ for some $\Upsilon_\eta > 0$ independent of τ. We apply the Taylor expansion and obtain

$$\begin{aligned}
\mathcal{L}_\eta^{\tau+1} &\le \mathcal{L}_\eta^\tau + \alpha_\tau \begin{pmatrix}\nabla_{\boldsymbol{z}}\mathcal{L}_\eta^\tau \\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta^\tau\end{pmatrix}^T \begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau \\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix} + \frac{\Upsilon_\eta\alpha_\tau^2}{2}\left\|\begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau \\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix}\right\|^2 \\
&\overset{(5.1)}{\le} \mathcal{L}_\eta^\tau + \alpha_\tau \begin{pmatrix}\nabla_{\boldsymbol{z}}\mathcal{L}_\eta^\tau \\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta^\tau\end{pmatrix}^T \begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau \\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix} + \frac{\Upsilon_\eta\alpha_\tau^2(1+\delta)^2}{2}\|(\Delta\boldsymbol{z}^\tau, \Delta\boldsymbol{\lambda}^\tau)\|^2 \\
&\overset{(3.2),\ \text{Lemma } 5.1}{\le} \mathcal{L}_\eta^\tau + \alpha_\tau \begin{pmatrix}\nabla_{\boldsymbol{z}}\mathcal{L}_\eta^\tau \\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta^\tau\end{pmatrix}^T \begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau \\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix} + 2\Upsilon_\eta\Upsilon^2\alpha_\tau^2\|\nabla\mathcal{L}^\tau\|^2 \\
&\overset{\text{Theorem } 5.1}{\le} \mathcal{L}_\eta^\tau + \alpha_\tau \begin{pmatrix}\nabla_{\boldsymbol{z}}\mathcal{L}_\eta^\tau \\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta^\tau\end{pmatrix}^T \begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau \\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix} - \frac{4\Upsilon_\eta\Upsilon^2}{\eta_2}\alpha_\tau^2 \begin{pmatrix}\nabla_{\boldsymbol{z}}\mathcal{L}_\eta^\tau \\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta^\tau\end{pmatrix}^T \begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau \\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix} \\
&= \mathcal{L}_\eta^\tau + \alpha_\tau \begin{pmatrix}\nabla_{\boldsymbol{z}}\mathcal{L}_\eta^\tau \\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta^\tau\end{pmatrix}^T \begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau \\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix} \cdot \left\{1 - \frac{4\Upsilon_\eta\Upsilon^2}{\eta_2}\alpha_\tau\right\},
\end{aligned}$$

where the third inequality also uses $\delta \le 1$. Thus, as long as

$$1 - \frac{4\Upsilon_\eta\Upsilon^2}{\eta_2}\alpha_\tau \ge \beta \Longleftrightarrow \alpha_\tau \le \frac{(1-\beta)\eta_2}{4\Upsilon_\eta\Upsilon^2},$$

the Armijo condition (3.5) is satisfied. Since the right hand side is independent of τ, we know $\alpha_\tau \ge \bar{\alpha}$ for some $\bar{\alpha} > 0$ when doing, for example, a backtracking line search. By (3.5) and Theorem 5.1,

$$\mathcal{L}_\eta^{\tau+1} \le \mathcal{L}_\eta^\tau - \frac{\eta_2\alpha_\tau\beta}{2}\|\nabla\mathcal{L}^\tau\|^2 \le \mathcal{L}_\eta^\tau - \frac{\eta_2\bar{\alpha}\beta}{2}\|\nabla\mathcal{L}^\tau\|^2.$$

Summing over τ, $\sum_{\tau=0}^\infty \|\nabla\mathcal{L}^\tau\|^2 \le \frac{2}{\eta_2\bar{\alpha}\beta}(\mathcal{L}_\eta^0 - \min_{\mathcal{Z}\times\Lambda}\mathcal{L}_\eta(\boldsymbol{z}, \boldsymbol{\lambda})) < \infty$. This completes the proof.

Appendix D: Proofs of results in section 6

D.1. Proof of Theorem 6.1 By (3.5), it suffices to show for large τ that

$$\mathcal{L}_\eta(\boldsymbol{z}^\tau+\tilde{\Delta}\boldsymbol{z}^\tau,\boldsymbol{\lambda}^\tau+\tilde{\Delta}\boldsymbol{\lambda}^\tau)\le\mathcal{L}_\eta^\tau+\beta\begin{pmatrix}\nabla_{\boldsymbol{z}}\mathcal{L}_\eta^\tau\\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta^\tau\end{pmatrix}^T\begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau\\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix}. \tag{D.1}$$

By the thrice continuous differentiability of $\{g_k,f_k\}$, we know $\nabla^2\mathcal{L}_\eta$ is continuous. Thus, we have the following Taylor expansion

$$\begin{aligned}&\mathcal{L}_\eta(\boldsymbol{z}^\tau+\tilde{\Delta}\boldsymbol{z}^\tau,\boldsymbol{\lambda}^\tau+\tilde{\Delta}\boldsymbol{\lambda}^\tau)\\ &\qquad\le\mathcal{L}_\eta^\tau+\begin{pmatrix}\nabla_{\boldsymbol{z}}\mathcal{L}_\eta^\tau\\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta^\tau\end{pmatrix}^T\begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau\\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix}+\frac{1}{2}\begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau\\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix}^T\nabla^2\mathcal{L}_\eta^\tau\begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau\\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix}+o(\|(\tilde{\Delta}\boldsymbol{z}^\tau,\tilde{\Delta}\boldsymbol{\lambda}^\tau)\|^2).\end{aligned} \tag{D.2}$$

By direct calculation, we have that

$$\begin{aligned}\nabla_{\boldsymbol{z}}^2\mathcal{L}_\eta=&H+\eta_1G^TG+\eta_2H^2+\eta_1\langle\nabla_{\boldsymbol{z}}G,f\rangle+\eta_2\langle\nabla_{\boldsymbol{z}}H,\nabla_{\boldsymbol{z}}\mathcal{L}\rangle,\\ \nabla_{\boldsymbol{z}\boldsymbol{\lambda}}\mathcal{L}_\eta=&G^T+\eta_2HG^T+\eta_2\langle\nabla_{\boldsymbol{\lambda}}H,\nabla_{\boldsymbol{z}}\mathcal{L}\rangle,\qquad\nabla_{\boldsymbol{\lambda}}^2\mathcal{L}_\eta=\eta_2GG^T,\end{aligned}$$

where for a function $a(x):\mathbb{R}^n\to\mathbb{R}^{m_1\times m_2}$ and a vector $b\in\mathbb{R}^{m_1}$, we let $\langle\nabla a(x),b\rangle:=\nabla^T(a^T(x)b)=\sum_{j=1}^{m_1}b_j\nabla^Ta_j(x)\in\mathbb{R}^{m_2\times n}$ for $a^T(x)=(a_1(x),\dots,a_{m_1}(x))$. Thus, we define

$$\mathcal{H}^\tau:=\begin{pmatrix}H^\tau+\eta_1(G^\tau)^TG^\tau+\eta_2(H^\tau)^2 & (I+\eta_2H^\tau)(G^\tau)^T\\ G^\tau(I+\eta_2H^\tau) & \eta_2G^\tau(G^\tau)^T\end{pmatrix},$$

apply $\|\nabla\mathcal{L}^\tau\|=\|(\nabla_{\boldsymbol{z}}\mathcal{L}^\tau,f^\tau)\|\to0$, and get

$$\left\|\nabla^2\mathcal{L}_\eta^\tau-\mathcal{H}^\tau\right\|=o(1). \tag{D.4}$$

Combining (D.1), (D.2) and (D.4), it suffices to show that

$$(1-\beta)\begin{pmatrix}\nabla_{\boldsymbol{z}}\mathcal{L}_\eta^\tau\\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta^\tau\end{pmatrix}^T\begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau\\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix}+\frac{1}{2}\begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau\\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix}^T\mathcal{H}^\tau\begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau\\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix}+o(\|(\tilde{\Delta}\boldsymbol{z}^\tau,\tilde{\Delta}\boldsymbol{\lambda}^\tau)\|^2)\le0. \tag{D.5}$$

We observe that

$$\begin{aligned}&\begin{pmatrix}\nabla_{\boldsymbol{z}}\mathcal{L}_\eta^\tau\\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta^\tau\end{pmatrix}^T\begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau\\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix}+\begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau\\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix}^T\mathcal{H}^\tau\begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau\\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix}\\ &\overset{\substack{(3.6)\\(3.2)}}{=}\begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau\\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix}^T\left\{\mathcal{H}^\tau-\begin{pmatrix}\hat{H}^\tau+\eta_1(G^\tau)^TG^\tau+\eta_2H^\tau\hat{H}^\tau & (I+\eta_2H^\tau)(G^\tau)^T\\ G^\tau(I+\eta_2\hat{H}^\tau) & \eta_2G^\tau(G^\tau)^T\end{pmatrix}\right\}\begin{pmatrix}\Delta\boldsymbol{z}^\tau\\ \Delta\boldsymbol{\lambda}^\tau\end{pmatrix}\\ &\qquad+\begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau\\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix}^T\mathcal{H}^\tau\begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau-\Delta\boldsymbol{z}^\tau\\ \tilde{\Delta}\boldsymbol{\lambda}^\tau-\Delta\boldsymbol{\lambda}^\tau\end{pmatrix}=:\mathcal{I}_3+\mathcal{I}_4.\end{aligned}$$

For term $\mathcal{I}_3$, we let $\Delta H^\tau=H^\tau-\hat{H}^\tau$ and have

$$\begin{aligned}\mathcal{I}_3=&\begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau\\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix}^T\begin{pmatrix}(I+\eta_2H^\tau)\Delta H^\tau & \mathbf{0}\\ \eta_2G^\tau\Delta H^\tau & \mathbf{0}\end{pmatrix}\begin{pmatrix}\Delta\boldsymbol{z}^\tau\\ \Delta\boldsymbol{\lambda}^\tau\end{pmatrix}\overset{(5.3)}{\le}\|(\tilde{\Delta}\boldsymbol{z}^\tau,\tilde{\Delta}\boldsymbol{\lambda}^\tau)\|\cdot(1+2\eta_2\Upsilon)\|\Delta H^\tau\Delta\boldsymbol{z}^\tau\|\\ \le&2(1+2\eta_2\Upsilon)\|(\Delta\boldsymbol{z}^\tau,\Delta\boldsymbol{\lambda}^\tau)\|o(\|\Delta\boldsymbol{z}^\tau\|)=o(\|(\Delta\boldsymbol{z}^\tau,\Delta\boldsymbol{\lambda}^\tau)\|^2),\end{aligned} \tag{D.6}$$

where the third inequality uses (5.1) and Assumption 6.1. For term $\mathcal{I}_4$, we apply Assumptions 4.3 and (5.3), and have

$$\begin{aligned}&\mathcal{I}_4\le\|(\tilde{\Delta}\boldsymbol{z}^\tau,\tilde{\Delta}\boldsymbol{\lambda}^\tau)\|\cdot\|\mathcal{H}^\tau\|\cdot\|(\Delta\boldsymbol{z}^\tau-\tilde{\Delta}\boldsymbol{z}^\tau,\Delta\boldsymbol{\lambda}^\tau-\tilde{\Delta}\boldsymbol{\lambda}^\tau)\|\\ &\overset{(5.1),(5.3)}{\le}2\delta\|(\Delta\boldsymbol{z}^\tau,\Delta\boldsymbol{\lambda}^\tau)\|^2\left\{\Upsilon+\eta_2\Upsilon^2+(\eta_1+\eta_2)\Upsilon^2+2(1+\eta_2\Upsilon)\Upsilon\right\}\overset{(C.9)}{\le}2.2\delta\eta_1\Upsilon^2\|(\Delta\boldsymbol{z}^\tau,\Delta\boldsymbol{\lambda}^\tau)\|^2.\end{aligned}$$

Combining the above display with (D.5), (D.6), and using $o(\|(\tilde{\Delta}\boldsymbol{z}^\tau, \tilde{\Delta}\boldsymbol{\lambda}^\tau)\|) \overset{(5.1)}{=} o(\|(\Delta\boldsymbol{z}^\tau, \Delta\boldsymbol{\lambda}^\tau)\|)$, it suffices to show that

$$(\frac{1}{2}-\beta)\begin{pmatrix}\nabla_{\boldsymbol{z}}\mathcal{L}_\eta^\tau\\ \nabla_{\boldsymbol{\lambda}}\mathcal{L}_\eta^\tau\end{pmatrix}^T\begin{pmatrix}\tilde{\Delta}\boldsymbol{z}^\tau\\ \tilde{\Delta}\boldsymbol{\lambda}^\tau\end{pmatrix}+1.1\delta\eta_1\Upsilon^2\|(\Delta\boldsymbol{z}^\tau,\Delta\boldsymbol{\lambda}^\tau)\|^2+o(\|(\Delta\boldsymbol{z}^\tau,\Delta\boldsymbol{\lambda}^\tau)\|^2)\le 0.$$

By (C.8), (C.10), (C.11) in the proof of Theorem 5.1, the above inequality holds if

$$(\frac{1}{2}-\beta)\left(\frac{\eta_2\gamma_G}{8}-1.1\delta\eta_1\Upsilon^2\right)\ge 1.1\delta\eta_1\Upsilon^2 \Longleftarrow \delta\le\frac{1/2-\beta}{3/2-\beta}\cdot\frac{\eta_2\gamma_G}{9\eta_1\Upsilon^2}.$$

This completes the proof.

D.2. Proof of Theorem 6.2 It suffices to show that the Newton system of $\mathcal{P}_\mu^i(\boldsymbol{d}_i^\tau)$ at $\mathcal{D}_i(\boldsymbol{z}^\tau,\boldsymbol{\lambda}^\tau)$ is the same as (3.9) with $\hat{H}_k^\tau = H_k^\tau$. We suppress the iteration index τ. The Newton system of $\mathcal{P}_\mu^i(\boldsymbol{d}_i^\tau)$ can be expressed as

$$\begin{pmatrix}H^{(i)} & (G^{(i)})^T\\ G^{(i)} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\Delta\boldsymbol{z}^{(i)}\\ \Delta\boldsymbol{\lambda}^{(i)}\end{pmatrix}=-\begin{pmatrix}\nabla_{\tilde{\boldsymbol{z}}_i}\mathcal{L}^{(i)}\\ \nabla_{\tilde{\boldsymbol{\lambda}}_i}\mathcal{L}^{(i)}\end{pmatrix},\tag{D.7}$$

where $\mathcal{L}^{(i)}$ is the Lagrangian function of $\mathcal{P}_\mu^i(\boldsymbol{d}_i^\tau)$, and $H^{(i)}=\nabla^2_{\tilde{\boldsymbol{z}}_i}\mathcal{L}^{(i)}$ and $G^{(i)}=\nabla_{\tilde{\boldsymbol{\lambda}}_i\tilde{\boldsymbol{z}}_i}\mathcal{L}^{(i)}$. By direct calculation and using the setup of $\boldsymbol{d}_i$ of procedure (a), we have

$$H^{(i)}=\operatorname{diag}(H_{m_1},\dots,H_{m_2-1},Q_{m_2}+\mu I),\tag{D.8a}$$

$$G^{(i)}=\begin{pmatrix}-A_{m_1} & -B_{m_1} & I & & & & \\ & & -A_{m_1+1} & -B_{m_1+1} & I & & \\ & & & & \ddots & \ddots & \ddots \\ & & & & & -A_{m_2-1} & -B_{m_2-1} \quad I\end{pmatrix},\tag{D.8b}$$

and

$$\nabla_{\tilde{\boldsymbol{z}}_i}\mathcal{L}^{(i)}=(\nabla_{\boldsymbol{z}_{m_1}}\mathcal{L};\dots;\nabla_{\boldsymbol{z}_{m_2-1}}\mathcal{L};\nabla_{\boldsymbol{x}_{m_2}}\mathcal{L}),\qquad \nabla_{\tilde{\boldsymbol{\lambda}}_i}\mathcal{L}^{(i)}=(\mathbf{0};\nabla_{\boldsymbol{\lambda}_{m_1+1}}\mathcal{L};\dots;\nabla_{\boldsymbol{\lambda}_{m_2}}\mathcal{L}).\tag{D.9}$$

Plugging (D.8) and (D.9) into (D.7), we observe that (D.7) is the same as $\mathcal{LP}_\mu^i(\boldsymbol{d}_i)$ in (3.9) with $\boldsymbol{d}_i=(\mathbf{0};\mathbf{0};\mathbf{0};\mathbf{0})$ and $\hat{H}_k=H_k$. Thus,

$$(\Delta\boldsymbol{z}^{(i)},\Delta\boldsymbol{\lambda}^{(i)})=(\tilde{\boldsymbol{w}}_i^\star(\boldsymbol{d}_i),\tilde{\boldsymbol{\zeta}}_i^\star(\boldsymbol{d}_i)).\tag{D.10}$$

Moreover, we denote by $(\boldsymbol{z}_s^{\tau+1},\boldsymbol{\lambda}_s^{\tau+1})$ and $(\boldsymbol{z}_f^{\tau+1},\boldsymbol{\lambda}_f^{\tau+1})$ the next iterate generated by the one-Newton-step Schwarz scheme and generated by FOTD, respectively. We have

$$\begin{aligned}(\boldsymbol{z}_s^{\tau+1},\boldsymbol{\lambda}_s^{\tau+1})&=\mathcal{C}\left(\left\{\mathcal{D}_i(\boldsymbol{z}^\tau,\boldsymbol{\lambda}^\tau)+(\Delta\boldsymbol{z}^{(i)},\Delta\boldsymbol{\lambda}^{(i)})\right\}_i\right)=\mathcal{C}\left(\{\mathcal{D}_i(\boldsymbol{z}^\tau,\boldsymbol{\lambda}^\tau)\}_i\right)+\mathcal{C}\left(\left\{(\Delta\boldsymbol{z}^{(i)},\Delta\boldsymbol{\lambda}^{(i)})\right\}_i\right)\\ &\overset{(D.10)}{=}\mathcal{C}\left(\{\mathcal{D}_i(\boldsymbol{z}^\tau,\boldsymbol{\lambda}^\tau)\}_i\right)+\mathcal{C}\left(\left\{(\tilde{\boldsymbol{w}}_i^\star(\boldsymbol{d}_i),\tilde{\boldsymbol{\zeta}}_i^\star(\boldsymbol{d}_i))\right\}_i\right)=\mathcal{C}\left(\{\mathcal{D}_i(\boldsymbol{z}^\tau,\boldsymbol{\lambda}^\tau)\}_i\right)+(\tilde{\Delta}\boldsymbol{z}^\tau,\tilde{\Delta}\boldsymbol{\lambda}^\tau)\\ &=(\boldsymbol{z}^\tau,\boldsymbol{\lambda}^\tau)+(\tilde{\Delta}\boldsymbol{z}^\tau,\tilde{\Delta}\boldsymbol{\lambda}^\tau)=(\boldsymbol{z}_f^{\tau+1},\boldsymbol{\lambda}_f^{\tau+1}),\end{aligned}$$

where the first, fourth and last equalities are due to the definitions of the Schwarz and the FOTD procedures; the second and fifth equalities are due to Definition 2.1. This completes the proof.

D.3. Proof of Lemma 6.1 Our proof relies on the KKT inverse structure in [35, Lemma 2]. We only show (6.4a), while (6.4b) holds by recalling that the last subproblem does not have boundary variables at the terminal stage N. For subproblem $i \in [M-2]$, we let $(\tilde{\boldsymbol{z}}_i^\star, \tilde{\boldsymbol{\lambda}}_i^\star) = \mathcal{D}_i(\boldsymbol{z}^\star, \boldsymbol{\lambda}^\star)$. Let C, ρ be the constants in Theorem 4.2, and let $C_1 = 9C\Upsilon^2/\gamma_G$. Then, under the assumptions and the setup of b in (6.3), $\delta = \gamma_G C_1 \rho^b/(9\Upsilon^2)$ satisfies (6.1). Suppose τ is large enough so that $\alpha_\tau = 1$. Borrowing the notation in (D.7)-(D.9), we let $\hat{H}^{(i)}$, $G^{(i)}$, $\nabla\mathcal{L}^{(i)}$ be the Hessian, Jacobian, and KKT residual vector of Problem (3.9) at the τ-th iterate $(\tilde{\boldsymbol{z}}_i^\tau, \tilde{\boldsymbol{\lambda}}_i^\tau)$. We consider the FOTD update:

$$\begin{aligned}\begin{pmatrix} \tilde{\boldsymbol{z}}_i^{\tau+1} - \tilde{\boldsymbol{z}}_i^\star \\ \tilde{\boldsymbol{\lambda}}_i^{\tau+1} - \tilde{\boldsymbol{\lambda}}_i^\star \end{pmatrix} &= \begin{pmatrix} \tilde{\boldsymbol{z}}_i^{\tau} - \tilde{\boldsymbol{z}}_i^\star \\ \tilde{\boldsymbol{\lambda}}_i^{\tau} - \tilde{\boldsymbol{\lambda}}_i^\star \end{pmatrix} + \begin{pmatrix} \tilde{\boldsymbol{w}}_i^\star(\boldsymbol{d}_i) \\ \tilde{\boldsymbol{\zeta}}_i^\star(\boldsymbol{d}_i) \end{pmatrix} \\ &\overset{\substack{(3.9)\\ \text{Theorem 6.2}}}{=} \begin{pmatrix} \hat{H}^{(i)} & (G^{(i)})^T \\ G^{(i)} & \mathbf{0} \end{pmatrix}^{-1} \left\{ \begin{pmatrix} \hat{H}^{(i)} & (G^{(i)})^T \\ G^{(i)} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{z}}_i^\tau - \tilde{\boldsymbol{z}}_i^\star \\ \tilde{\boldsymbol{\lambda}}_i^\tau - \tilde{\boldsymbol{\lambda}}_i^\star \end{pmatrix} - \begin{pmatrix} \nabla_{\tilde{\boldsymbol{z}}_i}\mathcal{L}^{(i)} \\ \nabla_{\tilde{\boldsymbol{\lambda}}_i}\mathcal{L}^{(i)} \end{pmatrix} \right\}. \end{aligned} \tag{D.11}$$

We define the KKT residual evaluated at the truncated full horizon solution $(\tilde{\boldsymbol{z}}_i^\star, \tilde{\boldsymbol{\lambda}}_i^\star)$ as

$$\begin{aligned} \nabla_{\tilde{\boldsymbol{z}}_i}\mathcal{L}^{(i),\star} &= (\nabla_{\boldsymbol{z}_{m_1}}\mathcal{L}^\star; \ldots; \nabla_{\boldsymbol{z}_{m_2-1}}\mathcal{L}^\star; \nabla_{\boldsymbol{x}_{m_2}}\tilde{\mathcal{L}}^\star), \\ \nabla_{\tilde{\boldsymbol{\lambda}}_i}\mathcal{L}^{(i),\star} &= (\boldsymbol{x}_{m_1}^\star - \boldsymbol{x}_{m_1}^\tau; \nabla_{\boldsymbol{\lambda}_{m_1+1}}\mathcal{L}^\star; \ldots; \nabla_{\boldsymbol{\lambda}_{m_2}}\mathcal{L}^\star), \end{aligned} \tag{D.12}$$

where $\nabla_{\boldsymbol{z}_{m_1:m_2-1}}\mathcal{L}^\star$ (similar for $\nabla_{\boldsymbol{\lambda}_{m_1+1:m_2}}\mathcal{L}^\star$) replaces the evaluation point $(\tilde{\boldsymbol{z}}_i^\tau, \tilde{\boldsymbol{\lambda}}_i^\tau)$ of the components $\nabla_{\boldsymbol{z}_{m_1:m_2-1}}\mathcal{L}$ of $\nabla_{\tilde{\boldsymbol{z}}_i}\mathcal{L}^{(i)}$ (cf. (D.9)) with $(\tilde{\boldsymbol{z}}_i^\star, \tilde{\boldsymbol{\lambda}}_i^\star)$, and

$$\nabla_{\boldsymbol{x}_{m_2}}\tilde{\mathcal{L}}^\star := \nabla_{\boldsymbol{x}_{m_2}} g_{m_2}(\boldsymbol{x}_{m_2}^\star, \boldsymbol{u}_{m_2}^\tau) + \boldsymbol{\lambda}_{m_2}^\star - A_{m_2}^T(\boldsymbol{x}_{m_2}^\star, \boldsymbol{u}_{m_2}^\tau)\boldsymbol{\lambda}_{m_2+1}^\tau + \mu(\boldsymbol{x}_{m_2}^\star - \boldsymbol{x}_{m_2}^\tau). \tag{D.13}$$

Clearly, if we change the evaluation point from $(\tilde{\boldsymbol{z}}_i^\star, \tilde{\boldsymbol{\lambda}}_i^\star)$ back to $(\tilde{\boldsymbol{z}}_i^\tau, \tilde{\boldsymbol{\lambda}}_i^\tau)$ in (D.12) and (D.13), then we get the vectors $\nabla_{\tilde{\boldsymbol{z}}_i}\mathcal{L}^{(i)}$ and $\nabla_{\tilde{\boldsymbol{\lambda}}_i}\mathcal{L}^{(i)}$ in (D.11). Moreover, for $0 \le \phi \le 1$, we let

$$\begin{aligned} \boldsymbol{u}_{m_2}^\star(\phi) &= \boldsymbol{u}_{m_2}^\star + \phi(\boldsymbol{u}_{m_2}^\tau - \boldsymbol{u}_{m_2}^\star), & \boldsymbol{\lambda}_{m_2+1}^\star(\phi) &= \boldsymbol{\lambda}_{m_2+1}^\star + \phi(\boldsymbol{\lambda}_{m_2+1}^\tau - \boldsymbol{\lambda}_{m_2+1}^\star), \\ \tilde{\boldsymbol{z}}_i^\star(\phi) &= \tilde{\boldsymbol{z}}_i^\star + \phi(\tilde{\boldsymbol{z}}_i^\tau - \tilde{\boldsymbol{z}}_i^\star), & \tilde{\boldsymbol{\lambda}}_i^\star(\phi) &= \tilde{\boldsymbol{\lambda}}_i^\star + \phi(\tilde{\boldsymbol{\lambda}}_i^\tau - \tilde{\boldsymbol{\lambda}}_i^\star), \end{aligned}$$

and let $H^{(i)}(\phi)$, $G^{(i)}(\phi)$ be $H^{(i)}$, $G^{(i)}$ (cf. (D.8)) evaluated at $(\tilde{\boldsymbol{z}}_i^\star(\phi), \tilde{\boldsymbol{\lambda}}_i^\star(\phi))$. Then, (D.11) implies

$$\begin{aligned} \begin{pmatrix} \tilde{\boldsymbol{z}}_i^{\tau+1} - \tilde{\boldsymbol{z}}_i^\star \\ \tilde{\boldsymbol{\lambda}}_i^{\tau+1} - \tilde{\boldsymbol{\lambda}}_i^\star \end{pmatrix} &= -\begin{pmatrix} \hat{H}^{(i)} & (G^{(i)})^T \\ G^{(i)} & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \nabla_{\tilde{\boldsymbol{z}}_i}\mathcal{L}^{(i),\star} \\ \nabla_{\tilde{\boldsymbol{\lambda}}_i}\mathcal{L}^{(i),\star} \end{pmatrix} \\ &\quad + \begin{pmatrix} \hat{H}^{(i)} & (G^{(i)})^T \\ G^{(i)} & \mathbf{0} \end{pmatrix}^{-1} \left\{ \begin{pmatrix} \hat{H}^{(i)} & (G^{(i)})^T \\ G^{(i)} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{z}}_i^\tau - \tilde{\boldsymbol{z}}_i^\star \\ \tilde{\boldsymbol{\lambda}}_i^\tau - \tilde{\boldsymbol{\lambda}}_i^\star \end{pmatrix} - \begin{pmatrix} \nabla_{\tilde{\boldsymbol{z}}_i}\mathcal{L}^{(i)} - \nabla_{\tilde{\boldsymbol{z}}_i}\mathcal{L}^{(i),\star} \\ \nabla_{\tilde{\boldsymbol{\lambda}}_i}\mathcal{L}^{(i)} - \nabla_{\tilde{\boldsymbol{\lambda}}_i}\mathcal{L}^{(i),\star} \end{pmatrix} \right\} \\ &= -\begin{pmatrix} \hat{H}^{(i)} & (G^{(i)})^T \\ G^{(i)} & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \nabla_{\tilde{\boldsymbol{z}}_i}\mathcal{L}^{(i),\star} \\ \nabla_{\tilde{\boldsymbol{\lambda}}_i}\mathcal{L}^{(i),\star} \end{pmatrix} \\ &\quad + \begin{pmatrix} \hat{H}^{(i)} & (G^{(i)})^T \\ G^{(i)} & \mathbf{0} \end{pmatrix}^{-1} \int_0^1 \begin{pmatrix} \hat{H}^{(i)} - H^{(i)}(\phi) & (G^{(i)})^T - (G^{(i)}(\phi))^T \\ G^{(i)} - G^{(i)}(\phi) & \mathbf{0} \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{z}}_i^\tau - \tilde{\boldsymbol{z}}_i^\star \\ \tilde{\boldsymbol{\lambda}}_i^\tau - \tilde{\boldsymbol{\lambda}}_i^\star \end{pmatrix} d\phi \\ &=: \mathcal{K}_i^{-1}\mathcal{J}_1 + \mathcal{K}_i^{-1}\mathcal{J}_2. \end{aligned}$$

To establish the stagewise error recursion, it suffices to establish the blockwise bound for the KKT inverse $\mathcal{K}_i^{-1}$ and the component-wise bound for vectors $\mathcal{J}_1$ and $\mathcal{J}_2$. The KKT inverse structure is given by [35, Lemma 2] (the conditions are satisfied by Corollary 4.1). We now deal with $\mathcal{J}_1$ and $\mathcal{J}_2$.

Term $\mathcal{J}_1$. By Theorem 2.1(i) (or checking the KKT conditions (A.2) in the appendix), we know $\nabla_{\boldsymbol{z}_{m_1:m_2-1}}\mathcal{L}^\star = \mathbf{0}$ and $\nabla_{\boldsymbol{\lambda}_{m_1+1:m_2}}\mathcal{L}^\star = \mathbf{0}$. Thus, only the last component of $\nabla_{\tilde{\boldsymbol{z}}_i}\mathcal{L}^{(i),\star}$ and the first

component of $\nabla_{\tilde{\boldsymbol{\lambda}}_i}\mathcal{L}^{(i),\star}$ are nonzero. The first component of $\nabla_{\tilde{\boldsymbol{\lambda}}_i}\mathcal{L}^{(i),\star}$ is trivially bounded by $\|\boldsymbol{x}^\star_{m_1} - \boldsymbol{x}^\tau_{m_1}\|$. For the last component of $\nabla_{\tilde{\boldsymbol{z}}_i}\mathcal{L}^{(i),\star}$, we have

$$\begin{aligned}
\nabla_{\boldsymbol{x}_{m_2}}\tilde{\mathcal{L}}^\star \overset{\text{(D.13)}}{=} & \big\{\nabla_{\boldsymbol{x}_{m_2}} g_{m_2}(\boldsymbol{x}^\star_{m_2}, \boldsymbol{u}^\tau_{m_2}) + \boldsymbol{\lambda}^\star_{m_2} - A^T_{m_2}(\boldsymbol{x}^\star_{m_2}, \boldsymbol{u}^\tau_{m_2})\boldsymbol{\lambda}^\tau_{m_2+1}\big\} + \mu(\boldsymbol{x}^\star_{m_2} - \boldsymbol{x}^\tau_{m_2}) \\
& \underbrace{- \big\{\nabla_{\boldsymbol{x}_{m_2}} g_{m_2}(\boldsymbol{x}^\star_{m_2}, \boldsymbol{u}^\star_{m_2}) + \boldsymbol{\lambda}^\star_{m_2} - A^T_{m_2}(\boldsymbol{x}^\star_{m_2}, \boldsymbol{u}^\star_{m_2})\boldsymbol{\lambda}^\star_{m_2+1}\big\}}_{\text{this is } \mathbf{0} \text{ by KKT conditions (cf. (A.2))}} \\
= & \int_0^1 \left(S^T_{m_2}(\boldsymbol{x}^\star_{m_2}, \boldsymbol{u}^\star_{m_2}(\phi), \boldsymbol{\lambda}^\star_{m_2+1}(\phi)) \;\; - A^T_{m_2}(\boldsymbol{x}^\star_{m_2}, \boldsymbol{u}^\star_{m_2}(\phi))\right) \begin{pmatrix} \boldsymbol{u}^\tau_{m_2} - \boldsymbol{u}^\star_{m_2} \\ \boldsymbol{\lambda}^\tau_{m_2+1} - \boldsymbol{\lambda}^\star_{m_2+1} \end{pmatrix} d\phi \\
& + \mu(\boldsymbol{x}^\star_{m_2} - \boldsymbol{x}^\tau_{m_2}) \\
\overset{\text{(5.3)}}{\leq} & (2\Upsilon + \mu)\|(\boldsymbol{z}^\tau_{m_2} - \boldsymbol{z}^\star_{m_2}; \boldsymbol{\lambda}^\tau_{m_2+1} - \boldsymbol{\lambda}^\star_{m_2+1})\|.
\end{aligned}$$

Thus, we have ($\preceq$ means component-wise $\leq$)

$$\mathcal{J}_1 \preceq \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \dfrac{(2\Upsilon+\mu)\left\|\begin{pmatrix} \boldsymbol{z}^\tau_{m_2} - \boldsymbol{z}^\star_{m_2} \\ \boldsymbol{\lambda}^\tau_{m_2+1} - \boldsymbol{\lambda}^\star_{m_2+1} \end{pmatrix}\right\|}{\|\boldsymbol{x}^\tau_{m_1} - \boldsymbol{x}^\star_{m_1}\|} \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad \mathcal{J}_2 \asymp \begin{pmatrix} o(\Psi^\tau_{m_1}) + O((\Psi^\tau_{m_1+1})^2) \\ \vdots \\ o(\Psi^\tau_{m_2-1}) + O((\Psi^\tau_{m_2})^2) \\ o(\Psi^\tau_{m_2}) \\ 0 \\ O((\Psi^\tau_{m_1})^2) \\ \vdots \\ O((\Psi^\tau_{m_2-1})^2) \end{pmatrix}. \tag{D.14}$$

Term $\mathcal{J}_2$. Applying Assumption 6.1 so that $\|\hat{H}^{(i)} - H^{(i)}\| = o(1)$, and using the Lipschitz continuity of $H^{(i)}$ and $\{A_k, B_k\}$ assumed by Assumption 6.2, we immediately obtain (D.14) for $\mathcal{J}_2$ ($\asymp$ means component-wise $=$).

Finally, we apply [35, Lemma 2] and know that, there exists a constant $\tilde{C} > 0$ independent of τ and algorithmic parameters β, η_1, η_2 (ρ is the same as Theorem 4.2), such that $\forall k \in [m_1, m_2]$,

$$\begin{aligned}
\|\tilde{\boldsymbol{z}}^{\tau+1}_{i,k} - \boldsymbol{z}^\star_k\| \leq & \tilde{C}\left\{\rho^{k-m_1}\|\boldsymbol{x}^\tau_{m_1} - \boldsymbol{x}^\star_{m_1}\| + \rho^{m_2-k}\left\|\begin{pmatrix} \boldsymbol{z}^\tau_{m_2} - \boldsymbol{z}^\star_{m_2} \\ \boldsymbol{\lambda}^\tau_{m_2+1} - \boldsymbol{\lambda}^\star_{m_2+1} \end{pmatrix}\right\|\right\} + \tilde{C}\sum_{j=m_1}^{m_2} \rho^{|k-j|}\cdot o(\Psi^\tau) \\
= & o(\Psi^\tau) + \tilde{C}\left\{\rho^{k-m_1}\|\boldsymbol{x}^\tau_{m_1} - \boldsymbol{x}^\star_{m_1}\| + \rho^{m_2-k}\left\|\begin{pmatrix} \boldsymbol{z}^\tau_{m_2} - \boldsymbol{z}^\star_{m_2} \\ \boldsymbol{\lambda}^\tau_{m_2+1} - \boldsymbol{\lambda}^\star_{m_2+1} \end{pmatrix}\right\|\right\},
\end{aligned} \tag{D.15}$$

where the second equality is due to $\sum_{j=m_1}^{m_2}\rho^{|k-j|} \leq 2\sum_{j=0}^{\infty}\rho^j < \infty$. The inequality (D.15) holds for $\|\tilde{\boldsymbol{\lambda}}^\tau_{i,k} - \boldsymbol{\lambda}^\star_k\|$ as well. Using $\Psi^{\tau+1}_k \leq \|\tilde{\boldsymbol{z}}^{\tau+1}_{i,k} - \boldsymbol{z}^\star_k\| + \|\tilde{\boldsymbol{\lambda}}^\tau_{i,k} - \boldsymbol{\lambda}^\star_k\|$, we rescale C_1 by $C_1 \leftarrow \max\{C_1, 2\tilde{C}\}$ and complete the proof.

References

[1] Barrows C, Hummon M, Jones W, Hale E (2014) Time domain partitioning of electricity production cost simulations. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), URL http://dx.doi.org/10.2172/1123223.

[2] Beccuti A, Geyer T, Morari M (2004) Temporal lagrangian decomposition of model predictive control for hybrid systems. *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, volume 3, 2509–2514, IEEE (IEEE), URL http://dx.doi.org/10.1109/cdc.2004.1428793.

[3] Bertsekas D (1996) *Constrained optimization and Lagrange multiplier methods* (Belmont, Mass: Athena Scientific), ISBN 1886529043, URL https://www.mit.edu/~dimitrib/Constrained-Opt.pdf.

[4] Bock H, Plitt K (1984) A multiple shooting algorithm for direct solution of optimal control problems. *IFAC Proceedings Volumes* 17(2):1603–1608, URL http://dx.doi.org/10.1016/s1474-6670(17)61205-9.

[5] Bock HG, Diehl MM, Leineweber DB, Schlöder JP (2000) A direct multiple shooting method for real-time optimization of nonlinear DAE processes. *Nonlinear Model Predictive Control*, 245–267 (Birkhäuser Basel), URL http://dx.doi.org/10.1007/978-3-0348-8407-5_14.

[6] Boggs PT, Tolle JW (1995) Sequential quadratic programming. *Acta Numerica* 4:1–51, URL http://dx.doi.org/10.1017/s0962492900002518.

[7] Boyd S (2010) *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*, volume 3 (Now Publishers), URL http://dx.doi.org/10.1561/2200000016.

[8] Buikis A (1999) A mathematical model for the heat treatment of glass fabric sheets. *IMA Journal of Management Mathematics* 10(1):55–86, URL http://dx.doi.org/10.1093/imaman/10.1.55.

[9] Buikis A, Kalis H (1999) The mathematical modelling of the nonlinear heat transport in thin plate. *Mathematical modelling and analysis* 4(1):44–50, URL http://dx.doi.org/10.1080/13926292.1999.9637109.

[10] Byrd RH, Nocedal J, Waltz RA (2006) Knitro: An integrated package for nonlinear optimization. *Nonconvex Optimization and Its Applications*, 35–59 (Springer US), URL http://dx.doi.org/10.1007/0-387-30065-1_4.

[11] Byrd RH, Schnabel RB, Shultz GA (1988) Parallel quasi-newton methods for unconstrained optimization. *Mathematical Programming* 42(1-3):273–306, URL http://dx.doi.org/10.1007/bf01589407.

[12] Cao T, Hall J, van de Geijn R (2002) Parallel cholesky factorization of a block tridiagonal matrix. *Proceedings. International Conference on Parallel Processing Workshop*, 327–335, IEEE (IEEE Comput. Soc), URL http://dx.doi.org/10.1109/icppw.2002.1039748.

[13] Chiang N, Petra CG, Zavala VM (2014) Structured nonconvex optimization of large-scale energy systems using PIPS-NLP. *2014 Power Systems Computation Conference*, 1–7, IEEE (IEEE), URL http://dx.doi.org/10.1109/pscc.2014.7038374.

[14] Curtis FE, Jiang H, Robinson DP (2014) An adaptive augmented lagrangian method for large-scale constrained optimization. *Mathematical Programming* 152(1-2):201–245, URL http://dx.doi.org/10.1007/s10107-014-0784-y.

[15] Dias BH, Tomim MA, Marcato ALM, Ramos TP, Brandi RBS, da Silva Junior IC, Filho JAP (2013) Parallel computing applied to the stochastic dynamic programming for long term operation planning of hydrothermal power systems. *European Journal of Operational Research* 229(1):212–222, ISSN 0377-2217, URL http://dx.doi.org/10.1016/j.ejor.2013.02.024.

[16] Diehl M, Bock H, Schlöder JP, Findeisen R, Nagy Z, Allgöwer F (2002) Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control* 12(4):577–585, URL http://dx.doi.org/10.1016/s0959-1524(01)00023-3.

[17] Diehl M, Ferreau HJ, Haverbeke N (2009) Efficient numerical methods for nonlinear MPC and moving horizon estimation. *Nonlinear Model Predictive Control*, 391–417 (Springer Berlin Heidelberg), URL http://dx.doi.org/10.1007/978-3-642-01094-1_32.

[18] Diehl M, Findeisen R, Bock H, Allgöwer F, Schlöder J (2005) Nominal stability of real-time iteration scheme for nonlinear model predictive control. *IEE Proceedings - Control Theory and Applications* 152(3):296–308, URL http://dx.doi.org/10.1049/ip-cta:20040008.

[19] Domahidi A, Zgraggen AU, Zeilinger MN, Morari M, Jones CN (2012) Efficient interior point methods for multistage problems arising in receding horizon control. *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 668–674, IEEE (IEEE), URL http://dx.doi.org/10.1109/cdc.2012.6426855.

[20] Dunn JC, Bertsekas DP (1989) Efficient dynamic programming implementations of newton's method for unconstrained optimal control problems. *Journal of Optimization Theory and Applications* 63(1):23–38, URL http://dx.doi.org/10.1007/bf00940728.

[21] Dunning I, Huchette J, Lubin M (2017) JuMP: A modeling language for mathematical optimization. *SIAM Review* 59(2):295–320, ISSN 0036-1445, URL http://dx.doi.org/10.1137/15m1020575.

[22] Fletcher R (1973) An exact penalty function for nonlinear programming with inequalities. *Mathematical Programming* 5(1):129–150, URL http://dx.doi.org/10.1007/bf01580117.

[23] Frasch JV, Gray A, Zanon M, Ferreau HJ, Sager S, Borrelli F, Diehl M (2013) An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles. *2013 European Control Conference (ECC)*, 4136–4141, IEEE (IEEE), URL http://dx.doi.org/10.23919/ecc.2013.6669836.

[24] Frasch JV, Sager S, Diehl M (2015) A parallel quadratic programming method for dynamic optimization problems. *Mathematical Programming Computation* 7(3):289–329, URL http://dx.doi.org/10.1007/s12532-015-0081-7.

[25] Frison G, Sorensen HHB, Dammann B, Jorgensen JB (2014) High-performance small-scale solvers for linear model predictive control. *2014 European Control Conference (ECC)*, 128–133, IEEE (IEEE), URL http://dx.doi.org/10.1109/ecc.2014.6862490.

[26] Glad T, Polak E (1979) A multiplier method with automatic limitation of penalty growth. *Mathematical Programming* 17(1):140–155, URL http://dx.doi.org/10.1007/bf01588240.

[27] Hong M, Luo ZQ, Razaviyayn M (2016) Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization* 26(1):337–364, URL http://dx.doi.org/10.1137/140990309.

[28] Keerthi SS, Gilbert EG (1988) Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations. *Journal of Optimization Theory and Applications* 57(2):265–293, URL http://dx.doi.org/10.1007/bf00938540.

[29] Kirches C, Wirsching L, Bock H, Schlöder J (2012) Efficient direct multiple shooting for nonlinear model predictive control on long horizons. *Journal of Process Control* 22(3):540–550, URL http://dx.doi.org/10.1016/j.jprocont.2012.01.008.

[30] Laine F, Tomlin C (2019) Parallelizing LQR computation through endpoint-explicit riccati recursion. *2019 IEEE 58th Conference on Decision and Control (CDC)*, 1395–1402, IEEE (IEEE), URL http://dx.doi.org/10.1109/cdc40024.2019.9029974.

[31] Lemaréchal C (2001) Lagrangian relaxation. *Lecture Notes in Computer Science*, 112–156 (Springer Berlin Heidelberg), URL http://dx.doi.org/10.1007/3-540-45586-8_4.

[32] Li W, Todorov E (2007) Iterative linearization methods for approximately optimal control and estimation of non-linear stochastic system. *International Journal of Control* 80(9):1439–1453, URL http://dx.doi.org/10.1080/00207170701364913.

[33] MathWorks (2022) Nonlinear heat transfer in thin plate URL https://www.mathworks.com/help/pde/ug/nonlinear-heat-transfer-in-a-thin-plate.html.

[34] Na S, Anitescu M (2020) Exponential decay in the sensitivity analysis of nonlinear dynamic programming. *SIAM Journal on Optimization* 30(2):1527–1554, URL http://dx.doi.org/10.1137/19m1265065.

[35] Na S, Anitescu M (2023) Superconvergence of online optimization for model predictive control. *IEEE Transactions on Automatic Control* 68(3):1383–1398, URL http://dx.doi.org/10.1109/tac.2022.3223323.

[36] Na S, Anitescu M, Kolar M (2022) An adaptive stochastic sequential quadratic programming with differentiable exact augmented lagrangians. *Mathematical Programming* URL http://dx.doi.org/10.1007/s10107-022-01846-z.

[37] Na S, Anitescu M, Kolar M (2023) Inequality constrained stochastic nonlinear optimization via active-set sequential quadratic programming. *Mathematical Programming* 1–75, URL http://dx.doi.org/10.1007/s10107-023-01935-7.

[38] Na S, Shin S, Anitescu M, Zavala VM (2022) On the convergence of overlapping Schwarz decomposition for nonlinear optimal control. *IEEE Transactions on Automatic Control* 1–16, URL http://dx.doi.org/10.1109/tac.2022.3194087.

[39] Nielsen I, Axehill D (2015) A parallel structure exploiting factorization algorithm with applications to model predictive control. *2015 54th IEEE Conference on Decision and Control (CDC)*, 3932–3938, IEEE (IEEE), URL http://dx.doi.org/10.1109/cdc.2015.7402830.

[40] Nielsen I, Axehill D (2016) An o (log n) parallel algorithm for newton step computations with applications to moving horizon estimation. *2016 European Control Conference (ECC)*, 1630–1636, IEEE (IEEE), URL http://dx.doi.org/10.1109/ecc.2016.7810524.

[41] Nocedal J, Wright SJ (2006) *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering (Springer New York), 2nd edition, ISBN 978-0387-30303-1; 0-387-30303-0, URL http://dx.doi.org/10.1007/978-0-387-40065-5.

[42] O'Donoghue B, Stathopoulos G, Boyd S (2013) A splitting method for optimal control. *IEEE Transactions on Control Systems Technology* 21(6):2432–2442, URL http://dx.doi.org/10.1109/TCST.2012.2231960.

[43] Petra CG, Schenk O, Lubin M, Gäertner K (2014) An augmented incomplete factorization approach for computing the schur complement in stochastic optimization. *SIAM Journal on Scientific Computing* 36(2):C139–C162, ISSN 1064-8275, URL http://dx.doi.org/10.1137/130908737.

[44] Pillo G (1994) Exact penalty methods. *Algorithms for Continuous Optimization*, 209–253 (Springer Netherlands), URL http://dx.doi.org/10.1007/978-94-009-0369-2_8.

[45] Pillo GD, Grippo L (1979) A new class of augmented lagrangians in nonlinear programming. *SIAM Journal on Control and Optimization* 17(5):618–628, URL http://dx.doi.org/10.1137/0317044.

[46] Pillo GD, Grippo L, Lampariello F (1980) A method for solving equality constrained optimization problems by unconstrained minimization. *Optimization Techniques*, 96–105 (Springer-Verlag), URL http://dx.doi.org/10.1007/bfb0006592.

[47] Pillo GD, Lucidi S (2002) An augmented lagrangian function with improved exactness properties. *SIAM Journal on Optimization* 12(2):376–406, URL http://dx.doi.org/10.1137/s1052623497321894.

[48] Roulet V, Srinivasa S, Fazel M, Harchaoui Z (2022) Iterative linear quadratic optimization for nonlinear control: Differentiable programming algorithmic templates. *arXiv preprint arXiv:2207.06362* URL https://arxiv.org/abs/2207.06362.

[49] Schittkowski K (1982) The nonlinear programming method of wilson, han, and powell with an augmented lagrangian type line search function. *Numerische Mathematik* 38(1):83–114, URL http://dx.doi.org/10.1007/bf01395810.

[50] Schäfer A, Kühl P, Diehl M, Schlöder J, Bock HG (2007) Fast reduced multiple shooting methods for nonlinear model predictive control. *Chemical Engineering and Processing: Process Intensification* 46(11):1200–1214, URL http://dx.doi.org/10.1016/j.cep.2006.06.024.

[51] Shin S, Anitescu M, Zavala VM (2022) Exponential decay of sensitivity in graph-structured nonlinear programs. *SIAM Journal on Optimization* 32(2):1156–1183, URL http://dx.doi.org/10.1137/21m1391079.

[52] Shin S, Faulwasser T, Zanon M, Zavala VM (2019) A parallel decomposition scheme for solving long-horizon optimal control problems. *IEEE 58th Conference on Decision and Control (CDC)* URL http://dx.doi.org/10.1109/cdc40024.2019.9030139.

[53] Shin S, Zavala VM (2021) Diffusing-horizon model predictive control. *IEEE Transactions on Automatic Control* 1–1, URL http://dx.doi.org/10.1109/tac.2021.3137100.

[54] Shin S, Zavala VM, Anitescu M (2020) Decentralized schemes with overlap for solving graph-structured optimization problems. *IEEE Transactions on Control of Network Systems* 7(3):1225–1236, URL http://dx.doi.org/10.1109/tcns.2020.2967805.

[55] Tassa Y, Erez T, Todorov E (2012) Synthesis and stabilization of complex behaviors through online trajectory optimization. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4906–4913 (IEEE), URL http://dx.doi.org/10.1109/iros.2012.6386025.

[56] Topaloglou N, Vladimirou H, Zenios SA (2008) A dynamic stochastic programming model for international portfolio management. *European Journal of Operational Research* 185(3):1501–1524, ISSN 0377-2217, URL http://dx.doi.org/10.1016/j.ejor.2005.07.035.

[57] Verschueren R, Zanon M, Quirynen R, Diehl M (2017) A sparsity preserving convexification procedure for indefinite quadratic programs arising in direct optimal control. *SIAM Journal on Optimization* 27(3):2085–2109, ISSN 1052-6234, URL http://dx.doi.org/10.1137/16m1081543.

[58] Wächter A, Biegler LT (2005) Line search filter methods for nonlinear programming: Local convergence. *SIAM Journal on Optimization* 16(1):32–48, ISSN 1052-6234, URL http://dx.doi.org/10.1137/s1052623403426544.

[59] Wächter A, Biegler LT (2005) Line search filter methods for nonlinear programming: Motivation and global convergence. *SIAM Journal on Optimization* 16(1):1–31, ISSN 1052-6234, URL http://dx.doi.org/10.1137/s1052623403426556.

[60] Wächter A, Biegler LT (2005) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming* 106(1):25–57, ISSN 0025-5610, URL http://dx.doi.org/10.1007/s10107-004-0559-y.

[61] Wan W, Biegler LT (2016) Structured regularization for barrier NLP solvers. *Computational Optimization and Applications* 66(3):401–424, ISSN 0926-6003, URL http://dx.doi.org/10.1007/s10589-016-9880-7.

[62] Wang Y, Yin W, Zeng J (2018) Global convergence of ADMM in nonconvex nonsmooth optimization. *Journal of Scientific Computing* 78(1):29–63, URL http://dx.doi.org/10.1007/s10915-018-0757-z.

[63] Wright SJ (1990) Solution of discrete-time optimal control problems on parallel computers. *Parallel Computing* 16(2-3):221–237, URL http://dx.doi.org/10.1016/0167-8191(90)90060-m.

[64] Wright SJ (1991) Parallel algorithms for banded linear systems. *SIAM Journal on Scientific and Statistical Computing* 12(4):824–842, URL http://dx.doi.org/10.1137/0912044.

[65] Xu W, Anitescu M (2018) Exponentially accurate temporal decomposition for long-horizon linear-quadratic dynamic optimization. *SIAM Journal on Optimization* 28(3):2541–2573, ISSN 1052-6234, URL http://dx.doi.org/10.1137/16m1081993.

[66] Xu W, Anitescu M (2019) Exponentially convergent receding horizon strategy for constrained optimal control. *Vietnam Journal of Mathematics* 47(4):897–929, ISSN 2305-221X, URL http://dx.doi.org/10.1007/s10013-019-00375-1.

[67] Zanelli A, Dinh QT, Diehl M (2020) Stability analysis of real-time methods for equality constrained NMPC. *IFAC-PapersOnLine* 53(2):6570–6576, URL http://dx.doi.org/10.1016/j.ifacol.2020.12.074.

[68] Zanon M, Frasch JV, Vukov M, Sager S, Diehl M (2014) Model predictive control of autonomous vehicles. *Optimization and Optimal Control in Automotive Systems*, 41–57 (Springer International Publishing), URL http://dx.doi.org/10.1007/978-3-319-05371-4_3.

[69] Zavala VM, Anitescu M (2014) Scalable nonlinear programming via exact differentiable penalty functions and trust-region newton methods. *SIAM Journal on Optimization* 24(1):528–558, ISSN 1052-6234, URL http://dx.doi.org/10.1137/120888181.