2025 IEEE 64th Conference on Decision and Control (CDC)
December 10-12, 2025. Rio de Janeiro, Brazil

A Time Splitting Based Optimization Method for Nonlinear MHE

Shuting Wu[†], Yifei Wang[†], Jingzhe Wang, Apostolos I. Rikos and Xu Du[*]

2025 IEEE 64th Conference on Decision and Control (CDC) | 979-8-3315-2627-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/CDC57313.2025.11312986

Abstract— **This paper presents computationally efficient algorithms for solving nonlinear Moving Horizon Estimation (MHE) problems, which face challenges due to the *curse of dimensionality*. Specifically, we first introduce a distributed reformulation utilizing a time-splitting technique. Leveraging this, we develop the Efficient Gauss-Newton Augmented Lagrangian Alternating Direction Inexact Newton (ALADIN) algorithm to improve efficiency. To address limited computational power in some sub-problem solvers, we propose the Efficient Sensitivity Assisted ALADIN, allowing inexact solutions without compromising performance. Additionally, we propose a Distributed Sequential Quadratic Programming (SQP) method for scenarios with no computational resources for sub-problems. Numerical experiments on a differential drive robot MHE problem demonstrate that our algorithms achieve both high accuracy and computational efficiency, meeting real-time requirements.**

I. Introduction

Moving Horizon Estimation (MHE) has attracted considerable interest for its applications in differential drive robots [1], unmanned aerial vehicles [2], and wireless communication [3]; a comprehensive overview is provided in [4]. Essentially, MHE is an optimization-based approach for estimating the states of dynamic systems within a moving time horizon, providing an effective framework for state estimation in nonlinear and constrained dynamic systems. Current MHE approaches mainly rely on centralized solvers, yet these methods become computationally prohibitive as estimation complexity and the length of time horizon increase - a challenge commonly described as the *curse of dimensionality*. To address this challenge, one promising approach is to reformulate MHE as a distributed optimization problem and adopt parallel algorithms for its solution. However, to the best of our knowledge, a suitable algorithm that efficiently solves distributed MHE has not yet been identified.

A natural approach for solving the distributed optimization reformulation of MHE is to adopt Augmented Lagrangian Alternating Direction Inexact Newton (ALADIN) [5], a distributed non-convex optimization algorithm known for integrating the advantages of Alternating Direction Method of Multipliers (ADMM) [6], [7] and Distributed Sequential Quadratic Programming (SQP) [8]. This motivation arises from ALADIN's demonstrated success in efficiently addressing Model Predictive Control problem (MPC) [9], [10], [11], [12], [13]- an optimization counterpart of MHE. ALADIN exhibits global convergence for convex problems and local convergence for non-convex problems [14], [15]–[17], with [18] establishing a global convergence theory for ALADIN in the context of non-convex problems. Typically, ALADIN solves sub-problems using an appropriate nonlinear programming (NLP) solver and coordinates information by solving a coupled quadratic programming (QP) problem [19]. However, directly applying standard ALADIN [20] to MHE remains computationally expensive due to the inherent coupled QP step required for coordinating distributed information, rendering it unsuitable for the real-time requirements of MHE. While a variant of ALADIN tailored for MPC [9] might be considered, it targets general objective functions (e.g., economic MPC [21]) rather than the specific least-squares objective of MHE. Although a variant of ALADIN, known as Gauss-Newton ALADIN [22], exists for handling least-squares objectives, it remains computationally inefficient due to the aforementioned coupled QP step. Thus, this gap motivates the following research question: *Can we develop computationally efficient variants of ALADIN specifically tailored to nonlinear MHE?*

Contributions

A. In this paper, we introduce a novel time-splitting-based optimization framework for solving nonlinear MHE problems efficiently while maintaining accuracy. We first revisit the nonlinear MHE formulation and propose a time-splitting-based distributed reformulation, extending the temporal decomposition concept originally developed for MPC [9]. Our reformulation partitions the time horizon into multiple independent sub-windows, significantly reducing sub-problems dimensionality.

B. Leveraging this distributed reformulation, we develop computationally efficient solutions within the ALADIN framework. Specifically, to eliminate the computational overhead associated with iterative QP solutions required in ALADIN, we first derive a closed-form solution for the

[*] Corresponding author.

[†] The first two authors contributed equally to this work.

Shuting Wu is with School of Mathematics and Statistics, North China University of Water Resources and Electric Power, Zhengzhou, China. Shuting Wu is also with the HNAS Institute of Mathematics, Henan Academy of Science. E-mail: wushuting0126@163.com.

Yifei Wang is with the Ningbo Artificial Intelligence Institute and the Department of Automation, Shanghai Jiao Tong University, Ningbo 315000, China, and also with the School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: yifeiw4ng@sjtu.edu.cn.

Jingzhe Wang is with School of Computing and Information, University of Pittsburgh, Pittsburgh, PA, USA. E-mail: jiw148@pitt.edu.

Apostolos I. Rikos and Xu Du are with the Artificial Intelligence Thrust of the Information Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China. Apostolos I. Rikos is also affiliated with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong, China. E-mails: apostolosr@hkust-gz.edu.cn; michaelxudu@hkust-gz.edu.cn.

The work of A.I.R. and X.D. are also supported by the Guangzhou-HKUST(GZ) Joint Funding Scheme (Grant No. 2025A03J3960).

QP step. Exploiting this closed-form solution, we propose Efficient Gauss-Newton ALADIN, an accelerated variant of Gauss-Newton ALADIN algorithm introduced in [22], which achieves computational efficiency.

C. Considering practical scenarios where sub-problem solvers possess limited computational power, we introduce Efficient Sensitivity Assisted ALADIN, inspired by [23], which allows the sub-problems step to be solved inexactly.

D. We further consider an extreme scenario wherein sub-problem solvers have no computational capability. Under this stringent condition, inspired by [24], we develop an Efficient Distributed SQP that entirely eliminates explicit sub-problem solving. Instead, it only evaluates first- and second-order information of local objectives.

E. We conducted numerical benchmarks on a practical nonlinear MHE problem involving the differential drive robots. The results demonstrate that our Efficient Distributed SQP achieves identical state estimation trajectories to those obtained by `CasADi` with `IPOPT` [1]. Moreover, all three proposed algorithms exhibit excellent stability in terms of iteration count and convergence precision. Notably, the fastest algorithm achieves high precision in a remarkably short time.

II. Fundamentals of the MHE

A. Discrete Control System

In control systems, dynamic behavior is typically modeled using discrete-time nonlinear equations, comprising state and output equations that characterize system evolution and observation relationships at time index n,

$$\begin{aligned} x_{n+1} &= f(x_n, u_n), \\ y_n &= h(x_n) + v_n. \end{aligned} \tag{1}$$

Here, $x_n \in \mathbb{R}^{|x_n|}$ denotes the system state, $u_n \in \mathbb{R}^{|u_n|}$ represents the control input, and $y_n \in \mathbb{R}^{|y_n|}$ stands for the measured output. Note that the measurement noise v_n follows a zero-mean Gaussian distribution, i.e., $v_n \sim \mathcal{N}(0, V)$, where V is a positive-definite covariance matrix. Furthermore, the nonlinear dynamics is defined by $f : \mathbb{R}^{|x_n|+|u_n|} \to \mathbb{R}^{|x_n|}$, and the nonlinear measurement function is expressed by $h : \mathbb{R}^{|x_n|} \to \mathbb{R}^{|y_n|}$, both of which are assumed to be twice continuously differentiable.

B. Basics of MHE

Based on (1), at each time step l, given a prediction horizon of length L, the following optimization problem represents a formulation of MHE (see [1]):

$$\begin{aligned} \min_{x,u} \ & \frac{1}{2}\|x_{l-L} - \hat{x}_{l-L}\|^2_{P^{-1}} + \frac{1}{2}\sum_{n=l-L}^{l} \|h(x_n) - y_n\|^2_{V^{-1}} \\ & + \frac{1}{2}\sum_{n=l-L}^{l-1} \|u_n - \hat{u}_n\|^2_{W^{-1}} + \frac{1}{2}\sum_{n=l-L}^{l-1} \|x_{n+1} - f(x_n, u_n)\|^2_{R^{-1}} . \end{aligned} \tag{2}$$

The optimization variable is defined as,

$$\begin{cases} x = \left(x_{l-L}^\top, x_{l-L+1}^\top, \ldots, x_l^\top\right)^\top, \\ u = \left(u_{l-L}^\top, u_{l-L+1}^\top, \ldots, u_{l-1}^\top\right)^\top, \end{cases} \tag{3}$$

where $\hat{x}_{l-L}$ represents the prior state estimate [4, Section 4.2], $P \in \mathbb{R}^{|x_n| \times |x_n|}$ denotes the covariance matrix associated with the initial state estimation error, $R \in \mathbb{R}^{|x_n| \times |x_n|}$ corresponds to the covariance matrix of the state noise, $V \in \mathbb{R}^{|y_n| \times |y_n|}$ describes the covariance matrix of the observation noise, and $W \in \mathbb{R}^{|u_n| \times |u_n|}$ characterizes the covariance matrix of the control input variations. In this expression, the optimization variables of (2) are x and u.

An alternative MHE formulation considers only x as the optimization variable. Although u still appears in the expressions, it is treated as a known constant. Based on this, the simplified optimization problem is formulated as follows[1] (see [28]):

$$\begin{aligned} & \min_{x} \frac{1}{2}\|x_{l-L} - \hat{x}_{l-L}\|^2_{P^{-1}} + \frac{1}{2}\sum_{n=l-L}^{l} \|h(x_n) - y_n\|^2_{V^{-1}} \\ & \text{s.t. } x_{n+1} = f(x_n, u_n), \qquad \forall n = l-L, \ldots, l-1. \end{aligned} \tag{4}$$

This paper focuses on the MHE optimization problem formulated in (4).

III. Distributed MHE Reformulation: A Time-Splitting-Based Approach

This section introduces a time-splitting-based distributed MHE framework built on (4). By partitioning the time horizon into multiple independent sub-windows, this approach significantly reduces the dimensionality of the sub-problems.

A. Components of the Time Splitting Reformulation

To mitigate computational complexity and enhance real-time performance in problem (4), the time window $[l-L, l]$ is divided into N consecutive sub-windows [9]. The first $(N-1)$ sub-windows each have a length of $t = \left\lfloor \frac{L}{N} \right\rfloor$, while the last sub-window has a length of $t_N = L - (N-1)t$, where $N, t, t_N \in \mathbb{N}_{>0}$. Accordingly, the time range for the i-th sub-window is given by $[l-L+(i-1)t, l-L+it]$, $(i = 1, 2, \ldots, N-1)$. For the last sub-window $(i = N)$, the time range is $[l-L+(N-1)t, l]$. Importantly, the auxiliary variable $z = \left((z_1)^\top, (z_2)^\top, \ldots, (z_N)^\top\right)^\top$ is introduced to represent the *boundary state* of each sub-window. Here, $z_i = \left((z_i^a)^\top, (z_i^b)^\top\right)^\top$ with z_i^a denoting the *initial state* of the i-th sub-window, defined as $z_i^a = x_{l-L+(i-1)t}$. In subsequent sections, $x_{l-L+(i-1)t}$ will be replaced by z_i^a. Meanwhile, z_i^b serves as a new auxiliary variable representing the *terminal state* of the i-th sub-window.

The optimization variable X_i associated with the local optimization problem for the i-th sub-window is defined as:

$$X_i = \left((z_i^a)^\top, (\tilde{x}_{(i)})^\top, (z_i^b)^\top\right)^\top, \quad X_i \in \mathbb{R}^{|X_i|},$$

where $\tilde{x}_{(i)}$ represents the *internal states* of the i-th sub-window, such that $\tilde{x}_{(i)} \in \mathbb{R}^{|\tilde{x}_{(i)}|}$, and is expressed as:

$$\tilde{x}_{(i)} = \begin{cases} \left((x_{l-L+(i-1)t+1})^\top, \ldots, (x_{l-L+it-1})^\top\right)^\top, & i=1, \ldots, N-1, \\ \left((x_{l-L+(N-1)t+1})^\top, \ldots, (x_{l-1})^\top\right)^\top, & i=N. \end{cases}$$

[1] For the convenience of the subsequent expressions, this paper studies MHE without inequality constraints. See [25]–[27] for a similar setting.

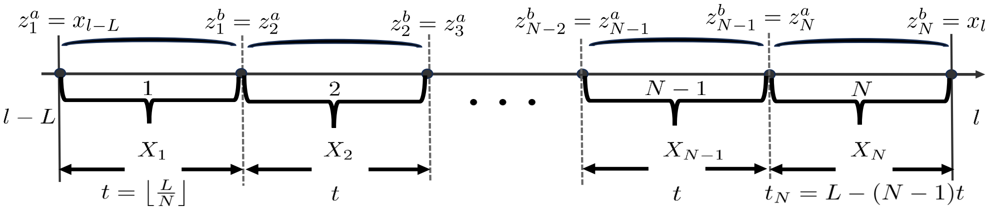

Fig. 1: The time-splitting-based MHE

With the above definitions, a schematic diagram of the *time-splitting-based MHE*, where $z_i^b = z_{i+1}^a$ is illustrated in Figure 1. Further details are provided in Section III-B.

The objective function for each sub-problem is represented by $J_i(X_i) : \mathbb{R}^{|X_i|} \to \mathbb{R}$, and the optimization problem for the i-th sub-window is formulated as follows, for $i = 1$ and $i = N$,

$$\begin{cases} J_1(X_1)=\frac{1}{2}\|z_1^a-\hat{x}_{l-L}\|_{P^{-1}}^2+\frac{1}{2}\sum_{j=l-L}^{l-L+t-1}\|h(x_j)-y_j\|_{V^{-1}}^2, \\ J_N(X_N) = \frac{1}{2}\sum_{j=l-L+(N-1)t}^{l}\|h(x_j)-y_j\|_{V^{-1}}^2, \end{cases} \tag{5}$$

for $i = 2, \cdots, N-1$,

$$J_i(X_i) = \frac{1}{2}\sum_{j=l-L+(i-1)t}^{l-L+it-1}\|h(x_j)-y_j\|_{V^{-1}}^2. \tag{6}$$

Analogous to the objective function formulation, the nonlinear dynamic equality constraints are partitioned into sub-vectors independently as follows, for $i = 1, \cdots, N-1$,

$$\mathcal{F}_i(X_i)=\begin{bmatrix} x_{l-L+(i-1)t+1}-f(z_i^a, u_{l-L+(i-1)t}) \\ x_{l-L+(i-1)t+2}-f(x_{l-L+(i-1)t+1}, u_{l-L+(i-1)t+1}) \\ \vdots \\ z_i^b-f(x_{l-L+it-1}, u_{l-L+it-1}) \end{bmatrix}, \tag{7}$$

for $i = N$:

$$\mathcal{F}_i(X_i)=\begin{bmatrix} x_{l-L+(i-1)t+1}-f(z_i^a, u_{l-L+(i-1)t}) \\ x_{l-L+(i-1)t+2}-f(x_{l-L+(i-1)t+1}, u_{l-L+(i-1)t+1}) \\ \vdots \\ x_l-f(x_{l-1}, u_{l-1}) \end{bmatrix}. \tag{8}$$

B. The Time Splitting Reformulation of MHE

Consequently, based on (5)-(8), the time-splitting-based formulation of MHE can be represented as:

$$\begin{aligned} \min_{\{X_i\}} \quad & \sum_{i=1}^{N} J_i(X_i) \\ \text{s.t.} \quad & \mathcal{F}_i(X_i) = 0 \quad |\mu_i, \quad \forall i = 1, \cdots, N, \\ & \sum_{i=1}^{N} A_i X_i = 0 \quad |\lambda. \end{aligned} \tag{9}$$

Here, μ_i represents the dual variable of the sub-constraint $\mathcal{F}_i$, where its dimension is given by,

$$|\mu_i| = \begin{cases} |X_i|(t-1), & i = 1, 2, \cdots, N-1, \\ |X_i|(t_N-1), & i = N, \end{cases}$$

while $\lambda \in \mathbb{R}^{(N-1)|z_1^b|}$ denotes the Lagrange multiplier corresponding to the coupling constraints. The coupling constraint matrix A_i is structurally defined as follows,

$$A_1 = \begin{bmatrix} \bar{\mathbf{0}} & \hat{\mathbf{0}} & I_{|z_1^b|} \\ \bar{\mathbf{0}} & \hat{\mathbf{0}} & \bar{\mathbf{0}} \\ \vdots & \vdots & \vdots \end{bmatrix}, A_N = \begin{bmatrix} \vdots & \vdots & \vdots \\ \bar{\mathbf{0}} & \mathbf{0}^{(N)} & \bar{\mathbf{0}} \\ -I_{|z_N^a|} & \mathbf{0}^{(N)} & \bar{\mathbf{0}} \end{bmatrix},$$

$$\tilde{A} = \begin{bmatrix} -I_{|z_i^a|} & \hat{\mathbf{0}} & \bar{\mathbf{0}} \\ \bar{\mathbf{0}} & \hat{\mathbf{0}} & I_{|z_i^b|} \end{bmatrix}, \forall i \in \{2, \cdots N-1\},$$

$$A_i = \begin{bmatrix} \mathbf{0}_{|X_i|\times(i-2)|z_1^b|}, & \tilde{A}^\top, & \mathbf{0}_{|X_i|\times(r-i|z_1^b|)} \end{bmatrix}^\top,$$

where, matrix $\bar{\mathbf{0}} = \mathbf{0}_{|z_1^b|\times|z_1^b|}$; $\hat{\mathbf{0}} = \mathbf{0}_{|z_1^b|\times|\tilde{x}_{(1)}|}$; $\mathbf{0}^{(N)} = \mathbf{0}_{|z_1^b|\times|\tilde{x}_{(N)}|}$; such that $A_1 \in \mathbb{R}^{r\times|X_1|}$, $A_i \in \mathbb{R}^{r\times|X_i|}$, $A_N \in \mathbb{R}^{r\times|X_N|}$, $\tilde{A} \in \mathbb{R}^{2|x_1^b|\times|X_i|}$. Note that $\sum_{i=1}^N A_i X_i = 0$ contains $z_i^b = z_{i+1}^a$, for $i = 1, \cdots, N-1$.

IV. Distributed Optimization Algorithms

This section is dedicated to developing efficient solutions within the ALADIN framework to address the time-splitting reformulation of MHE (9). Initially, we propose an efficient approach for solving coupled QP, which is integrated into the ALADIN framework. Subsequently, based on the aforementioned efficient approach, three ALADIN variants are proposed to reduce the computational burden of the standard ALADIN [20]. In this section, $(\cdot)^+$ denotes the value after the update, whereas $(\cdot)^-$ represents the value before the update.

A. An Efficient Method for Solving Coupled QP

Before introducing our algorithm for solving problem (9), we first introduce an efficient method for solving the strongly convex QP (10) below with coupling constraints:

$$\begin{aligned} \min_{\{\Delta X_i\}} \quad & \sum_{i=1}^{N} \frac{1}{2}\Delta X_i^\top H_i \Delta X_i + g_i^\top \Delta X_i \\ \text{s.t.} \quad & C_i \Delta X_i = 0 \qquad |\mu_i, \quad \forall i = 1, \cdots, N, \\ & \sum_{i=1}^{N} A_i (X_i^+ + \Delta X_i) = 0 \; |\lambda. \end{aligned} \tag{10}$$

Theorem 1 *(Efficient QP) Let the locally linear independence constraint qualification (LICQ) be satisfied for problem* (10)*, ensuring the linear independence of* C_i*s and* A_i*s for every* $i = 1, 2, \cdots, N$*. Let the locally second-order sufficient condition (SOSC) [8] be satisfied, i.e.,* $H_i \succ 0, \forall i$*. Also, let us assume the existence of a unique global optimal* *solution for problem* (10)*. Solving problem* (10) *is equivalent to evaluating the values of* λ, μ_i *and* ΔX_i *as follows,*

$$\left\{\begin{aligned} \lambda &= \left(\sum_{i=1}^{N} G_i - Q_i R_i^{-1} Q_i^\top\right)^{-1} p, \\ \mu_i &= -\, R_i^{-1}\left(C_i H_i^{-1} g_i + Q_i^\top \lambda\right), \\ \Delta X_i &= -\, H_i^{-1}\left(g_i + C_i^\top \mu_i + A_i^\top \lambda\right), \end{aligned}\right. \tag{11}$$

where,

$$\left\{\begin{aligned} G_i &= A_i H_i^{-1} A_i^\top, \\ Q_i &= A_i H_i^{-1} C_i^\top, \\ R_i &= C_i H_i^{-1} C_i^\top, \end{aligned}\right. \quad \left\{\begin{aligned} q &= \sum_{i=1}^{N} \left(Q_i R_i^{-1} C_i - A_i\right) H_i^{-1} g_i, \\ p &= \sum_{i=1}^{N} A_i X_i^+ + q. \end{aligned}\right. \tag{12}$$

Proof. See Appendix I. ■

As an extension of Theorem 1, we propose the closed-form solution

$$\left\{\begin{aligned} \lambda &= \left(\sum_{i=1}^{N} G_i - Q_i R_i^{-1} Q_i^\top\right)^{-1}\left(p - \sum_{i=1}^{N} Q_i R_i^{-1} D_i\right), \\ \mu_i &= -R_i^{-1}\left(C_i H_i^{-1} g_i + Q_i^\top \lambda - D_i\right), \\ \Delta X_i &= -H_i^{-1}\left(g_i + C_i^\top \mu_i + A_i^\top \lambda\right). \end{aligned}\right. \tag{13}$$

of the following problem,

$$\begin{aligned} \min_{\{\Delta X_i\}} \quad & \sum_{i=1}^{N} \frac{1}{2} \Delta X_i^\top H_i \Delta X_i + g_i^\top \Delta X_i \\ \text{s.t.} \quad & D_i + C_i \Delta X_i = 0 \qquad |\mu_i, \quad \forall i = 1, \cdots, N, \\ & \sum_{i=1}^{N} A_i (X_i^+ + \Delta X_i) = 0 \;|\lambda, \end{aligned} \tag{14}$$

where $D_i \in \mathbb{R}^{|\mu_i|}$ are given constant matrices. Note that, equations (11) and (13) will be integrated into our proposed algorithms. Due to space limitations, details are omitted here.

B. Algorithm Development

Based on [22], Section IV-B.1 introduces an efficient variant of Gauss-Newton ALADIN. Section IV-B.2 presents an inexact update version of ALADIN, inspired by [23]. Finally, Section IV-B.3 explores an ALADIN variant in which sub-problems are not locally optimized (drawing inspiration from [24]).

1) ***Efficient Gauss-Newton ALADIN****:* The objective function $J_i(X_i)$ in (9) is formulated as a nonlinear least-squares optimization problem, where the full vector-valued measurement function $\mathcal{H}_i(X_i)$ is introduced:

$$\mathcal{H}_i(X_i) = \begin{pmatrix} P^{-\frac{1}{2}} \left(z_1^a - \hat{x}_{l-L}\right)_{i=1} \\ V^{-\frac{1}{2}} \left(h(x_j) - y_j\right)_{j \in \mathcal{I}_i} \\ V^{-\frac{1}{2}} \left(h(x_l) - y_l\right)_{i=N} \end{pmatrix}, \tag{15}$$

where, $\mathcal{I}_i = \{l - L + (i-1)t, \cdots, l - L + it - 1\}$. Consequently, the objective function $J_i(X_i)$ of the sub-problems can be expressed as $J_i(X_i) = \frac{1}{2}\|\mathcal{H}_i(X_i)\|^2$.

Efficient Gauss-Newton ALADIN is presented in Algorithm 1. Similar to Gauss-Newton ALADIN [22], it alternates between solving sub-problems in parallel at the sub-nodes and coordinating via the coupled QP (10). Further, Algorithm 1 replaces the coupled QP with (11), thereby accelerating computation. During each iteration, Step 1) solves the NLP sub-problems (16) in parallel using any NLP solver. In Step 2), each sub-node performs sensitivity analysis based on its local solution, computing the gradient g_i and H_i at each local node according to the optimal solution X_i^+, see (17). These results are then transmitted to the central node. After gathering the sensitivity data from all sub-nodes, the central node updates the global dual variable λ in Step 4) using equation (19). The updated λ is subsequently broadcast to the sub-nodes, allowing each sub-node to locally update the primal variables according to equation (20). This process is repeated until convergence.

Algorithm 1 Efficient Gauss-Newton ALADIN

```
Initialization: Initial guess of dual variable $\lambda$ and primal variables $\{Y_i\}$, $\forall i$, choose $\rho > 0$.
Output: Optimal solution $\{Y_i^\star\}$.
Repeat:
    1) Paralleled solve local NLP:
        $X_i{}^+ = \arg\min_{X_i} \frac{1}{2}\|\mathcal{H}_i(X_i)\|^2 + \lambda^\top A_i X_i + \frac{\rho}{2}\|X_i - Y_i^-\|^2$
        $\text{s.t.} \quad \mathcal{F}_i(X_i) = 0.$    (16)
    2) Evaluate local variables and sensitivity matrix from $X_i^+$:
        $b_i = \mathcal{H}_i(X_i^+)$,
        $B_i = \nabla \mathcal{H}_i(X_i^+)^\top$,    (17)
        $C_i = \nabla \mathcal{F}_i(X_i^+)$.
    3) Assemble gradient and Hessian:
        $g_i = B_i b_i, \quad H_i = B_i B_i^\top.$    (18)
    4) Update and broadcast the global dual variable $\lambda$:
        $\lambda = \left(\sum_{i=1}^{N} G_i - Q_i R_i^{-1} Q_i^\top\right)^{-1} p.$    (19)
    5) Paralleled update local primal and dual variables:
        $\mu_i = -\, R_i^{-1}(C_i H_i^{-1} g_i + Q_i^\top \lambda)$,
        $Y_i^+ = X_i^+ - H_i^{-1}(g_i + C_i^\top \mu_i + A_i^\top \lambda)$.    (20)
```

Note that Algorithm 1 is specifically tailored for least-squares problems. To extend its applicability and further reduce overall computational time, we propose two additional ALADIN variants designed for broader problem classes.

2) ***Efficient Sensitivity Assisted ALADIN****:* Inspired by [23], we propose Efficient Sensitivity Assisted ALADIN (Algorithm 2) by leveraging the sensitivity of NLP parameters.

The augmented Lagrangian function for each sub-problem of problem (9) is expressed as

$$\mathcal{L}_i = J_i(X_i) + \lambda^\top A_i (X_i - Y_i) + \frac{\rho}{2}\|X_i - Y_i\|^2 + \mu_i^\top \mathcal{F}_i(X_i). \tag{21}$$

Following the notation in [23, IV.C], we define $s_i(\xi_i) = (X_i(\xi_i)^\top, \mu_i(\xi_i)^\top)^\top$ for notational convenience, where $\xi_i = (Y_i^\top, \lambda^\top)^\top$. The Karush-Kuhn-Tucker (KKT) conditions for the constrained sub-problems can be further expressed as,

$$\varphi_i(s_i(\xi_i^-), \xi_i^-) = \begin{bmatrix} \nabla_{X_i} \mathcal{L}_i(s_i(\xi_i^-)) \\ \mathcal{F}_i(X_i^-) \end{bmatrix} = 0, \tag{22}$$

where higher-order terms in the linearization of the solution manifold are neglected, the update for the sub-problems of Algorithm 2 is as follows,

$$s_i^+(\xi_i) = s_i(\xi_i^-) - \mathcal{M}_i^{-1}\mathcal{N}_i\left(\xi_i - \xi_i^-\right), \tag{23}$$

where $\mathcal{M}_i = \frac{\partial \varphi_i}{\partial s_i}$, $\mathcal{N}_i = \frac{\partial \varphi_i}{\partial \xi_i}$.

Utilizing a tangent predictor, the approximate solutions of the sub-problems at subsequent iterations can be efficiently estimated. Unlike the linearized ALADIN method [18, Equation (12), Appendix A], which linearizes the objective function around the current iteration point, this approach instead focuses on linearizing the solution manifold in the vicinity of the parameters.

```
Algorithm 2 Efficient Sensitivity Assisted ALADIN
Initialization: Initial guess of dual variable $\lambda$, $\mu_i$, primal variables $\{Y_i = X_i\}$, $\forall i$ and parameter $\xi_0 = ((Y_i)^\top, \lambda^\top)^\top$, choose $\rho > 0$.
Output: Optimal solution $\{Y_i^\star\}$.
Repeat:
    1) Evaluate gradient, Hessian and sensitivity matrix from $X_i$:
         $g_i = \nabla J_i(X_i)$,
         $H_i \approx \nabla^2(J_i(X_i) + \mu_i^\top \mathcal{F}_i(X_i)) + \rho I$,      (24)
         $C_i = \nabla \mathcal{F}_i(X_i)$,
         $D_i = \mathcal{F}_i(X_i)$.
    2) Update and the global dual variable $\lambda$ as
         $\lambda = \left(\sum_{i=1}^{N} G_i - Q_i R_i^{-1} Q_i^\top\right)^{-1}\left(p - \sum_{i=1}^{N} Q_i R_i^{-1} D_i\right)$.      (25)
    3) Paralleled update $\mu_i$s and $Y_i^+$s as
         $\hat{\mu}_i = -R_i^{-1}\left(C_i H_i^{-1} g_i + Q_i^\top \lambda - D_i\right)$,
         $Y_i^+ = X_i - H_i^{-1}\left(g_i + C_i^\top \hat{\mu}_i + A_i^\top \lambda\right)$.      (26)
    4) Collect parameter $\xi_i = \left((Y_i^+)^\top, \lambda^\top\right)^\top$, compute $\mathcal{M}_i, \mathcal{N}_i$ in parallel, and then solve local NLP with (23) [2].
    5) Extract $X_i^+$ from $s_i^+$: $s_i^+ = ((X_i^+)^\top, \mu_i^\top)^\top$.
```

In Algorithm 2, inspired by (13), the central node updates the global dual variable according to equation (25), incorporating local information from equation (24). Each node then concurrently updates its local dual variable $\hat{\mu}_i$ and primal variable Y_i^+ via equation (26). Next, each node updates s_i^+ using (23). This process iterates until convergence.

*3) **Efficient Distributed SQP**:* Building on the approach proposed in Decentralized SQP [24], we propose Efficient Distributed SQP (Algorithm 3). Unlike Algorithm 1 and 2, Algorithm 3 solves problem (9) by bypassing the resolution of sub-problems. Moreover, instead of solving the coupled QP (14) via an inner-level ADMM [24], Algorithm 3 updates the global dual variable λ, the local variables μ_i and ΔX_i according to the closed-form given by (13).

```
Algorithm 3 Efficient Distributed SQP
Initialization: Initial guess of dual variable $\lambda$ and primal variables $\{Y_i\}$, $\forall i$, choose $\rho > 0$.
Output: Optimal solution $\{Y_i^\star\}$.
Repeat:
    1) Locally update gradient, Hessian and sensitivity matrix from $Y_i^-$:
         $g_i = \nabla J_i(Y_i^-)$,
         $H_i \approx \nabla^2(J_i(Y_i^-) + \mu_i^\top \mathcal{F}_i(Y_i^-)) + \rho I$,      (27)
         $C_i = \nabla \mathcal{F}_i(Y_i^-)$,
         $D_i = \mathcal{F}_i(Y_i^-)$.
    2) Update and the global dual as equation (25).
    3) Update $\mu_i$s and $Y_i^+$s as
         $\mu_i = -R_i^{-1}(C_i H_i^{-1} g_i + Q_i^\top \lambda - D_i)$,
         $Y_i^+ = Y_i^- - H_i^{-1}(g_i + C_i^\top \mu_i + A_i^\top \lambda)$.      (28)
```

C. Convergence Analysis

We now examine the variations and convergence properties of the algorithms presented. Specifically, compared to Gauss-Newton ALADIN in [22], Algorithm 1 incorporates the closed-form expression given in (10) (as detailed in equations (19) and (20)). A comprehensive convergence analysis for Gauss-Newton ALADIN is provided in [22, Theorem 1]. Algorithm 2 features a local update step inspired from the Sensitivity-Assisted ADMM [29]. The corresponding convergence analysis will be included in the extended version of this work. The convergence analysis for Algorithm 3 is derived from [24, Theorem 1], and thus, is omitted for brevity.

V. Numerical Experiment

In this section, we apply the three proposed algorithms to a practical MHE problem, known as the differential drive robots problem (see [1]). As demonstrated in [5], the MPC problem locally satisfies the conditions of Theorem 1. Given that the MHE problem is shown to be the dual of the MPC problem in [30, Section 2.2], it follows that the practical MHE problem also locally satisfies Theorem 1. The following MHE problem involves three state variables, $x = (\phi, \psi, \theta)^\top$, which represent the lateral position ϕ, longitudinal position ψ, and orientation angle θ. Additionally, two control inputs, $u = (v, \omega)^\top$, are considered, where v denotes the linear velocity and ω the angular velocity. The observation vector $y = (r, \alpha)^\top$ consists of the relative range r and bearing α. Given x, u, y and a sampling time of $T = 0.2s$, the dynamics of the MHE system and the observer model are formulated as follows, in contrast to equation (1):

$$f(x_n, u_n) = \begin{bmatrix}\phi_n\\ \psi_n\\ \theta_n\end{bmatrix} + T\begin{bmatrix}v_n\cos\theta_n\\ v_n\sin\theta_n\\ \omega_n\end{bmatrix}, y_n = \begin{bmatrix}r\\ \alpha\end{bmatrix} = \begin{bmatrix}\sqrt{\phi_n^2 + \psi_n^2}\\ \arctan\left(\frac{\psi_n}{\phi_n}\right)\end{bmatrix} + \begin{bmatrix}\nu_r\\ \nu_\alpha\end{bmatrix},$$

[2] The update of local primal variables can optionally consist of two phases [23, Algorithm 1]: update using (23) when the KKT condition is almost satisfied; otherwise, update using (16).

where ν_r and ν_α denote Gaussian noise, with $\nu_r \sim \mathcal{N}(0, \sigma_r^2)$ and $\nu_\alpha \sim \mathcal{N}(0, \sigma_\alpha^2)$.

The code implementation in this paper is based on [1]. The experimental setup adopts a prediction horizon $L = 25$, and the initial states $x_0 = (0.1, 0.1, 0.0)^\top$ define the initial position and orientation of the robot. In the implementation, the state trajectories $x^* = (\phi^*, \psi^*, \theta^*)^\top$ are generated via MPC under the same control model as [1]. Notably, the primal variables are initialized to $(\phi^*, \psi^*, 0)^\top$. In the numerical implementation of Algorithms 1, the penalty parameter is set to $\rho = 25$, while for Algorithm 2 and 3, $\rho = 10^3$. The dual variables λ and μ are initialized to zero. All simulations were conducted using `Casadi-3.6.6` [31] with `IPOPT` in `MATLAB R2024a` on a Windows 11 system, equipped with a 2.1 GHz AMD Ryzen 5 4600U processor and 16GB of RAM.

Figure 2 compares the state trajectories obtained from centralized and distributed solvers for the MHE problem with $N = 4$. The results indicate that the proposed distributed MHE framework generates estimates nearly identical to those of the centralized baseline.

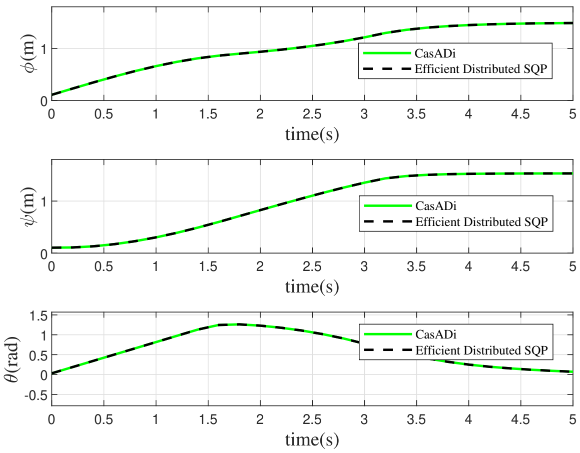

Fig. 2: Comparison of state estimation trajectories: centralized solver (`CasADi`) vs. Efficient Distributed SQP.

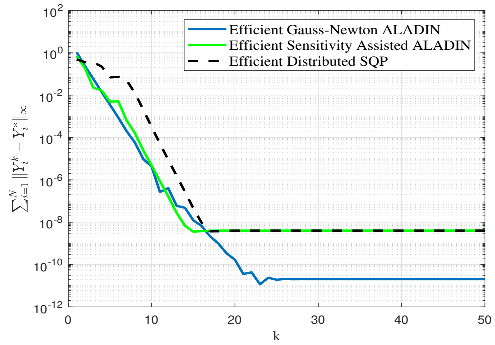

Fig. 3: Convergence comparison among Algorithms 1-3.

N	Algorithm 1	Algorithm 2	Algorithm 3	QP-CasADi
3	2.83	8.51	0.0183	1.60
4	2.84	8.76	0.0184	1.71
5	**2.04**	**8.08**	**0.0155**	**1.57**
6	3.05	11.82	0.0186	1.66

Table. 1: Total CPU time $[s]$ for different algorithms over N sub-windows (measured as the time for 50 iterations of each algorithm).

Figure 3 illustrates the convergence behavior of Algorithm 1–3 with $N = 4$, all exhibiting linear convergence. Notably, all three algorithms achieve an accuracy of 10^{-8} within 30 iterations, highlighting their computational efficiency. In particular, Algorithm 2 leverages `CasADi` to compute the exact solution in its first iteration, following the recommendations in [23, Algorithm 1].

Table 1 summarizes the total CPU time of the three proposed efficient ALADIN variants as a function of the number of sub-windows N. Notably, existing time-splitting-based MPC studies lack theoretical analysis on the relationship between the number of sub-windows N and computational time. By leveraging equations (11) and (13), we establish that the optimal number of sub-windows follows the asymptotic relation $N^* \approx \sqrt{L}$, where L denotes the total horizon length. For brevity, the detailed derivation will be provided in an extended version of this work. In our experiment, setting $L = 25$ yields an optimal sub-window count of $N^* = 5$. For comparison, we introduce **QP-CasADi**, which replaces Steps 2) and 3) of Algorithm 3 with `CasADi`-based QP solvers. The results demonstrate that across all four algorithmic structures, the configuration with $N = 5$ consistently achieves the lowest computational time. The centralized problem (4) was solved in approximately 0.0516 seconds using `CasADi`. Notably, although Algorithm 2 requires a longer total CPU time than Algorithm 1, its sub-problems solutions does not rely on existing solvers, making it particularly suitable for scenarios with limited computational resources at sub-nodes. As we know, [9] has already integrated the ALADIN algorithm into the ACADO Toolkit for experimental comparisons in the context of MPC problems. The present study could potentially be extended to conduct similar experiments.

VI. Conclusion

This paper introduces three computationally efficient distributed optimization algorithms for nonlinear MHE problems, considering sub-problem solver capabilities. We propose a distributed MHE reformulation using a time-splitting strategy and develop new solutions within the ALADIN algorithmic family. By utilizing a closed-form solution for large-scale coupled QP, these algorithms significantly reduce computational time, enabling real-time applications. Numerical experiments on an MHE problem with differential drive robots demonstrate superior convergence and efficiency. Future work will focus on enhancing ALADIN by accelerating matrix updates in Algorithm 2 and adaptively prioritizing critical sub-problems, as well as exploring the algorithms' applicability to larger-scale systems.

Appendix I
Proof of Theorem 1

The augmented Lagrangian function for problem (10) is defined as:

$$\begin{aligned}\mathcal{L}(\Delta X_i, \mu_i, \lambda)=&\sum_{i=1}^{N}\left(\frac{1}{2}\Delta X_i^\top H_i \Delta X_i + g_i^\top \Delta X_i\right)\\ &+\sum_{i=1}^{N}\mu_i^\top C_i \Delta X_i + \lambda^\top \sum_{i=1}^{N} A_i(X_i^+ + \Delta X_i).\end{aligned} \tag{29}$$

From (29), the KKT system of problem (10) is given by:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \Delta X_i} = H_i \Delta X_i + g_i + C_i^\top \mu_i + A_i^\top \lambda = 0,\\ \frac{\partial \mathcal{L}}{\partial \mu_i} = C_i \Delta X_i = 0,\\ \frac{\partial \mathcal{L}}{\partial \lambda} = \sum_{i=1}^{N} A_i(\Delta X_i + X_i^+) = 0. \end{cases} \tag{30}$$

From the first condition $\frac{\partial \mathcal{L}}{\partial \Delta X_i} = 0$ in equation (30), the following expression is derived:

$$\Delta X_i = -H_i^{-1}(g_i + C_i^\top \mu_i + A_i^\top \lambda). \tag{31}$$

When equation (31) is substituted into the second equation of (30), the resulting equation is expressed as:

$$\mu_i = -R_i^{-1}(C_i H_i^{-1} g_i + Q_i^\top \lambda). \tag{32}$$

Next, by substituting (32) into (31) and the third equation of (30), the following result is derived:

$$\sum_{i=1}^{N} G_i \lambda = p + \sum_{i=1}^{N} Q_i R_i^{-1} Q_i^\top \lambda.$$

Through further simplification, the solution for λ is obtained as equation (19). Subsequently, the local dual variable μ_i is computed by equation (32) using the previously computed λ. Finally, the local primal variable increment ΔX_i is calculated using equation (31) based on the obtained μ_i and λ. Consequently, problem (10) has been successfully solved.

References

[1] M. G. Mehrez, "MPC and MHE implementation in Matlab using CasADi," *github*, 2022.

[2] B. Wang, Z. Ma, S. Lai, and L. Zhao, "Neural moving horizon estimation for robust flight control," *IEEE Transactions on Robotics*, vol. 40, pp. 639–659, 2023.

[3] L. Zou, Z. Wang, B. Shen, and H. Dong, "Moving horizon estimation over relay channels: Dealing with packet losses," *Automatica*, vol. 155, p. 111079, 2023.

[4] J. B. Rawlings, D. Q. Mayne, M. Diehl *et al.*, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[5] D. Kouzoupis, R. Quirynen, B. Houska, and M. Diehl, "A block based ALADIN scheme for highly parallelizable direct optimal control," in *American Control Conference*, 2016, pp. 1124–1129.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[7] Z. Lin, H. Li, and C. Fang, *Alternating direction method of multipliers for machine learning*. Springer, 2022.

[8] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, New York, 2006.

[9] Y. Jiang, C. N. Jones, and B. Houska, "A time splitting based real-time iteration scheme for nonlinear MPC," in *IEEE Conference on Decision and Control*, 2019, pp. 2350–2355.

[10] J. Shi, Y. Jiang, J. Oravec, and B. Houska, "Parallel MPC for linear systems with state and input constraints," 2022. [Online]. Available: https://arxiv.org/abs/2207.00037

[11] J. Püttschneider, J. Golembiewski, N. A. Wagner, C. Wietfeld, and T. Faulwasser, "Towards event-triggered NMPC for efficient 6G communications: Experimental results and open problems," *arXiv preprint arXiv:2409.18589*, 2024.

[12] G. Stomberg, A. Engelmann, M. Diehl, and T. Faulwasser, "Decentralized real-time iterations for distributed nonlinear model predictive control," *arXiv preprint arXiv:2401.14898*, 2024.

[13] G. Stomberg, M. Raetsch, A. Engelmann, and T. Faulwasser, "Large problems are not necessarily hard: A case study on distributed NMPC paying off," *arXiv preprint arXiv:2411.05627*, 2024.

[14] B. Houska and Y. Jiang. Springer, 2021, ch. Distributed Optimization and Control with ALADIN, pp. 135–163.

[15] X. Du and J. Wang, "Consensus ALADIN: A framework for distributed optimization and its application in federated learning," *arXiv preprint arXiv:2306.05662*, 2023.

[16] ——, "Distributed consensus optimization with consensus ALADIN," in *American Control Conference*, 2025 (accepted for publication).

[17] X. Du, X. Zhou, S. Zhu, and A. I. Rikos, "Convergence theory of flexible ALADIN for distributed optimization," in *European Control Conference (ECC)*, 2025 (accepted for publication).

[18] X. Du, J. Wang, X. Zhou, and Y. Mao, "A bi-level globalization strategy for non-convex consensus ADMM and ALADIN," 2023.

[19] Y. Wang, S. Wu, G. Yang, J. Chu, A. I. Rikos, and X. Du, "Aladin-β: A distributed optimization algorithm for solving MPCC problems," in *Conference on Decision and Control (CDC)*, 2025 (accepted for publication).

[20] B. Houska, J. Frasch, and M. Diehl, "An augmented Lagrangian based algorithm for distributed nonconvex optimization," *SIAM Journal on Optimization*, vol. 26, no. 2, pp. 1101–1127, 2016.

[21] M. Diehl, R. Amrit, and J. B. Rawlings, "A Lyapunov function for economic optimizing model predictive control," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 703–707, March 2011. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5672577

[22] X. Du, A. Engelmann, Y. Jiang, T. Faulwasser, and B. Houska, "Distributed state estimation for AC power systems using Gauss-Newton ALADIN," in *IEEE Conference on Decision and Control*, 2019, pp. 1919–1924.

[23] D. Krishnamoorthy and V. Kungurtsev, "A sensitivity assisted alternating directions method of multipliers for distributed optimization," in *IEEE Conference on Decision and Control*, 2022, pp. 295–300.

[24] G. Stomberg, A. Engelmann, and T. Faulwasser, "Decentralized nonconvex optimization via bi-level SQP and ADMM," in *IEEE Conference on Decision and Control*, 2022, pp. 273–278.

[25] L. Simpson, J. Asprion, S. Muntwiler, J. Köhler, and M. Diehl, "Parallelizable parametric nonlinear system identification via tuning of a moving horizon state estimator," 2024. [Online]. Available: https://arxiv.org/abs/2403.17858

[26] M. Diehl, H. J. Ferreau, and N. Haverbeke, "Efficient numerical methods for nonlinear mpc and moving horizon estimation," *Nonlinear model predictive control: towards new challenging applications*, pp. 391–417, 2009.

[27] K. Baumgärtner, R. Reiter, and M. Diehl, "Moving horizon estimation with adaptive regularization for ill-posed state and parameter estimation problems," in *IEEE Conference on Decision and Control*, 2022, pp. 2165–2171.

[28] A. Wynn, M. Vukov, and M. Diehl, "Convergence guarantees for moving horizon estimation based on the real-time iteration scheme," *IEEE Transactions on Automatic Control*, vol. 59, no. 8, pp. 2215–2221, 2014.

[29] D. Krishnamoorthy and V. Kungurtsev, "Sensitivity assisted alternating directions method of multipliers for distributed optimization and statistical learning," *arXiv preprint arXiv:2009.05845*, 2020.

[30] M. Diehl, H. J. Ferreau, and N. Haverbeke, "Efficient numerical methods for nonlinear MPC and moving horizon estimation," in *Nonlinear model predictive control*, ser. Lecture Notes in Control and Information Sciences, L. Magni, M. Raimondo, and F. Allgöwer, Eds. Springer, 2009, vol. 384, pp. 391–417.

[31] J. A. E. Andersson, J. Gillis, , G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, Mar 2019.