

Lightweight Real-Time ALADIN for Distributed Optimization

Yifei Wang* Xuhui Feng** Shimin Pan** Liangfan Zhu**
Xu Du[†]*** Apostolos I. Rikos****

* *The Ningbo Artificial Intelligence Institute and the Department of
Automation, Shanghai Jiao Tong University, Ningbo, China
The School of Automation and Intelligent Sensing, Shanghai Jiao Tong
University, Shanghai, China (e-mail: yifeiw4ng@sjtu.edu.cn)*

** *Huawei Technologies Co., Ltd., Shanghai, China (e-mail:
{fengxuhui, panshimin1, zhuliangfan}@huawei.com)*

*** *The Artificial Intelligence Thrust of the Information Hub, The
Hong Kong University of Science and Technology (Guangzhou),
Guangzhou, China (e-mail:*

{michaelxudu,apostolosr}@hkust-gz.edu.cn).

**** *The Department of Computer Science and Engineering, The Hong
Kong University of Science and Technology, Clear Water Bay, Hong
Kong, China.*

Abstract: This paper presents a real-time computational framework for multi-node distributed optimization by extending the Augmented Lagrangian Alternating Direction Inexact Newton (ALADIN) algorithm. Our approach integrates adjoint sequential quadratic programming (SQP) techniques to enable efficient approximation of Jacobian information within the ALADIN embedded quadratic program, thereby reducing communication overhead. Furthermore, to decrease computational complexity, we design an event-triggered update strategy that avoids updating Hessian and Jacobian matrices at every iteration. The proposed method achieves local convergence and enhanced communication efficiency, making it well suited for time-critical applications. Numerical experiments demonstrate that our approach achieves competitive performance while exhibiting superior computational efficiency in real-time scenarios, validating its practical applicability for time-sensitive distributed optimization challenges.

Keywords: Distributed optimization, ALADIN, Adjoint SQP

1. INTRODUCTION

Distributed optimization has attracted substantial attention in recent years. This trend is driven by progress in federated learning (Ren et al. (2026); Du and Wang (2023)), power systems (Du et al. (2019); Lanza et al. (2025)), and model predictive control (MPC) (Stomberg et al. (2025); Kouzoupis et al. (2016)). Studies in these fields consistently show that the curse of dimensionality becomes severe as the dimension of the optimization variables grows. As a result, centralized methods for large-scale problems become computationally infeasible. Distributed optimization algorithms are therefore viewed as an effective means to alleviate the curse of dimensionality.

Distributed optimization comprises a family of computational methods in which nodes iteratively solve local subproblems. During the iteration process, nodes exchange information with neighbors or with a central coordinator.

The exchanged information may include first and second-order information, or local primal variables. These approaches are commonly classified into primal and dual decomposition frameworks (Doostmohammadian et al. (2025)). In dual decomposition methods, communication includes not only the information exchanged in primal decomposition but also dual variables associated with coupling constraints, which can enhance convergence behavior. A well-known algorithm in this category is the Alternating Direction Method of Multipliers (ADMM) (Boyd et al. (2011)), which distributes the Augmented Lagrangian Method and provides rigorous convergence guarantees for convex problems. However, ADMM often suffers from slow convergence and requires restrictive assumptions in nonconvex settings, as shown in Hong et al. (2016). Recent research has introduced the Augmented Lagrangian Alternating Direction Inexact Newton algorithm (ALADIN) (Houska and Shi (2022); Du and Wang (2025)). ALADIN incorporates sequential quadratic programming (SQP) into an ADMM-type framework, which yields accelerated convergence for convex problems and ensures local convergence for nonconvex cases. Building on these properties, this paper aims to explore the application of

* [†] Corresponding author

**The work of X.D. and A.I.R. was supported by the Guangzhou-HKUST(GZ) Joint Funding Scheme (Grant No. 2025A03J3960). The work of A.I.R. was also supported by the Guangdong Provincial Project (Grant No. 2024QN11G109).

ALADIN to distributed nonconvex optimization in real-time environments.

Existing Literature. ALADIN was originally developed for distributed sensor network localization (Houska et al. (2016)) and MPC (Kouzoupis et al. (2016)), both of which involve distributed nonconvex optimization problems. Subsequent studies (Engelmann et al. (2019); Du et al. (2019)) demonstrated its effectiveness in distributed alternating current power flow optimization. All of these applications fall within the category of resource allocation problems. More recently, Du and Wang (2025) introduced a consensus-oriented variant, termed Consensus ALADIN, which achieves improved computational efficiency for consensus optimization. To the best of our knowledge, the paper (Wang et al. (2025)) was the first to extend the application of ALADIN to non-smooth, non-convex constraint optimization problems. Regarding convergence properties, ALADIN ensures local convergence for nonconvex problems under standard assumptions. These assumptions include the second order sufficient condition (SOSC) and the linear independent constraint qualification (LICQ). For convex problems, global convergence can be achieved when the local Hessian is approximated by constant matrices. Recent theoretical advances (Du and Wang (2023, 2025); Du et al. (2025)) established sufficient conditions for global linear convergence of ALADIN in convex settings. Moreover, (Fedorová, 2025, Chapter 5) derived explicit linear convergence rates under stronger assumptions. For comprehensive surveys of ALADIN, we refer interested readers to Lanza et al. (2025).

However, a key limitation of the ALADIN framework lies in its reliance on solving a large-scale constrained coupled quadratic program (QP) for inter-node coordination. This step introduces high communication overhead and significant computational cost, mainly due to the construction and direct solution of the coupled QP (Du et al., 2019, IV.C). Recent work (Wu et al. (2025)) derived a closed-form expression for the constrained coupled QP. This result decouples the updates of three variable sets: (i) the dual variables associated with coupling constraints, (ii) the dual variables related to local constraints, (iii) the primal variables. Despite this simplification, constructing the coordination QP still requires each node to transmit substantial local information to the coordinator when local constraints are present. This information includes first and second-order information of the local objectives, the constraint Jacobians, which results in significant communication overhead. Moreover, the variable updates in the closed-form representation still rely on matrix inversions. These inversions become computationally expensive in high-dimensional settings. Although existing Hessian approximation techniques, such as the Broyden–Fletcher–Goldfarb–Shanno (BFGS)-based method in Du and Wang (2025), can avoid the direct transmission of Hessian information, they are only applicable to unconstrained subproblems and do not provide a general mechanism for handling constrained cases. These limitations reveal two major challenges for ALADIN: (i) excessive information transmission and the absence of effective Hessian and Jacobian recovery for constrained local subproblems, and (ii) considerable matrix inversion operations required to solve the large-scale coupled QP.

Main Contributions. Motivated by the aforementioned challenges, we propose a novel real-time optimization framework based on ALADIN, termed *Efficient Adjoint BFGS ALADIN*. Our approach combines the BFGS-ALADIN technique (Du and Wang (2025)) with adjoint SQP methods (Diehl et al. (2010); Quirynen et al. (2018); Feng et al. (2020); Hespanhol and Quirynen (2021)), enabling the coordinator to reconstruct both local Hessian and Jacobian information without explicit transmission from the nodes. Furthermore, inspired by Houska and Shi (2022), we incorporate an event-triggered mechanism that adaptively determines when to update these local approximations during the iterative process. To the best of our knowledge, this is the first work that employs ALADIN within a framework that simultaneously (i) reconstructs both Hessian and Jacobian information at the coordinator, and (ii) integrates an event-triggered update mechanism for real-time efficiency. Numerical results demonstrate the superior performance of the proposed algorithm compared to existing benchmarks.

Notation. In this paper, the superscript $[k]$ denotes the iteration index, while the notation $(\psi|\phi)$ indicates the dual multiplier ψ associated with the corresponding constraint ϕ .

2. PROBLEM FORMULATION

This paper considers a resource allocation problem involving N nodes. Each node i has a decision variable $x_i \in \mathbb{R}^{n_i}$, a local objective function $f_i(x_i) : \mathbb{R}^{n_i} \rightarrow \mathbb{R}$ and local constraints $g_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{c_i}$. The problem is formulated as follows:

$$\min_{\substack{x_i \in \mathbb{R}^{n_i} \\ \forall i \in \{1, \dots, N\}}} \sum_{i=1}^N f_i(x_i) \quad \text{s.t.} \quad \begin{cases} \sum_{i=1}^N A_i x_i = b, \\ g_i(x_i) = 0 \quad \forall i \in \{1, \dots, N\}. \end{cases} \quad (1)$$

Here, the nodes' variables are coupled through an affine constraint $\sum_{i=1}^N A_i x_i = b$, where $A_i \in \mathbb{R}^{m \times n_i}$ and $b \in \mathbb{R}^m$.

In this paper, we adopt the following assumptions that are essential for our subsequent development.

Assumption 1. SOSC Satisfaction. For each node, the local cost function f_i is closed, proper, twice continuously differentiable, and satisfies SOSC at a local minimizer x^* of problem (1), i.e. $\nabla^2 f(x^*) \succ 0$. Additionally, for every node i , the local constraint function g_i is assumed to be closed, proper, and second-order continuously differentiable.

Assumption 2. LICQ Satisfaction. At the local minimizer x^* of problem (1), the Jacobian of all equality constraints is full row rank. This ensures that the KKT system associated with problem (1) is regular.

Assumptions 1 and 2 guarantee that x^* is a regular Karush-Kuhn-Tucker (KKT) point of problem (1), which forms the basis for the local convergence of ALADIN-type methods for nonconvex problems. For further details, we refer to Houska et al. (2016); Kouzoupis et al. (2016).

3. PRELIMINARIES: EFFICIENT ALADIN

The Lagrangian function of problem (1) is defined as follows:

$$\mathcal{L}(x, \lambda) = \left(\sum_{i=1}^N f_i(x_i) + \mu_i^\top g_i(x_i) \right) + \lambda^\top \left(\sum_{i=1}^N A_i x_i - b \right). \quad (2)$$

Here, $x = [x_1^\top, x_2^\top, \dots, x_N^\top]^\top$, where μ_i denotes the local dual variable associated with the constraint $g_i(x_i) = 0$, and λ indicates the coupled dual variable corresponding to the coupling constraint.

Based on (2), in our previous work Wu et al. (2025)), we developed an *Efficient Gauss-Newton ALADIN* framework for solving moving horizon estimation (MHE) problems. The transformation from the distributed formulation in (1) to the algorithmic structure in (3) follows this framework, and the detailed derivation of the augmented Lagrangian decomposition and the associated coordination QP can be found in Wu et al. (2025). This approach reformulates the centralized MHE problem as a distributed optimization problem in the form of (1) (details can be found in Section 5.1). The algorithmic structure is presented below:

$$\begin{cases} (x_i^{[k+1]}, \mu_i^{[k+1]}) = \arg \min_{x_i} f_i(x_i) + (\lambda^{[k]})^\top x_i + \frac{\rho}{2} \|x_i - y_i^{[k]}\|^2, \\ \text{s.t. } g_i(x_i) = 0 \mid \mu_i, \quad \forall i \in \mathcal{V}, \\ H_i \approx \nabla^2(f_i(x_i^{[k+1]}) + (\mu_i^{[k+1]})^\top g_i(x_i^{[k+1]})) \succ 0, \\ v_i = \nabla f_i(x_i^{[k+1]}), \quad C_i = \nabla g_i(x_i^{[k+1]}), \\ \lambda^{[k+1]} = \left(\sum_{i=1}^N G_i - Q_i R_i^{-1} Q_i^\top \right)^{-1} p, \\ \tilde{\mu}_i = -R_i^{-1} (C_i H_i^{-1} v_i + Q_i^\top \lambda^{[k+1]}), \\ \Delta x_i = -H_i^{-1} (\nabla f_i(x_i^{[k+1]}) + C_i^\top \tilde{\mu}_i + A_i^\top \lambda^{[k+1]}), \\ y^{[k+1]} = x^{[k+1]} + \Delta x. \end{cases} \quad (3)$$

where the auxiliary matrices and vectors are defined as:

$$\begin{cases} G_i = A_i H_i^{-1} A_i^\top, \\ Q_i = A_i H_i^{-1} C_i^\top, \\ R_i = C_i H_i^{-1} C_i^\top, \end{cases} \quad \text{and} \quad \begin{cases} q = \sum_{i=1}^N (Q_i R_i^{-1} C_i - A_i) H_i^{-1} v_i, \\ p = \sum_{i=1}^N A_i x_i^{[k+1]} + q. \end{cases} \quad (4)$$

Moreover, let $y = [y_1^\top, y_2^\top, \dots, y_N^\top]^\top$ denote the stacked global variables and $\Delta x = [\Delta x_1^\top, \Delta x_2^\top, \dots, \Delta x_N^\top]^\top$ their corresponding increments (as shown in (3)). For each subproblem, H_i , C_i , and v_i denote the Gauss-Newton Hessian approximation, the constraint Jacobian, and the local objective gradient, respectively. The variables $\tilde{\mu}_i$ represent the dual multipliers of the linearized local constraints in the coordination QP and are used only to compute the primal search directions, unlike the local multipliers $\mu_i^{[k+1]}$ obtained from the local subproblems.

The algorithm consists of three main steps: first, each node solves its local augmented Lagrangian subproblem with parameter ρ and the previously updated primal variable $y_i^{[k]}$ received from all nodes; second, based on the updated local solution $x_i^{[k+1]}$, the gradient $v_i = \nabla f_i(x_i^{[k+1]})$, Hessian H_i of each objective function f_i , and the local constraint Jacobian C_i of $g_i(x_i^{[k+1]})$ are evaluated; third, the coordinator updates $\lambda^{[k+1]}$, $\tilde{\mu}_i$, and Δx_i for all $i \in \{1, 2, \dots, N\}$ using the closed-form expression for the affine-coupled constrained QP proposed in (Wu et al., 2025, Appendix I):

$$\begin{aligned} \min_{\substack{\Delta x_i \in \mathbb{R}^{n_i} \\ \forall i \in \{1, \dots, N\}}} & \sum_{i=1}^N \frac{1}{2} \Delta x_i^\top H_i \Delta x_i + v_i^\top \Delta x_i \\ \text{s.t.} & \sum_{i=1}^N A_i (\Delta x_i + x_i^{[k+1]}) = b \mid \lambda \\ & C_i \Delta x_i = 0, \mid \tilde{\mu}_i \quad \forall i \in \{1, \dots, N\}, \end{aligned} \quad (5)$$

and broadcast $(\lambda^{[k+1]}, y^{[k+1]})$ to all nodes, as derived in the last step of (3). The above steps are repeated until convergence. Note that Assumptions 1 and 2 guarantee the existence and uniqueness of the solution to the coupled QP (5).

4. LIGHTWEIGHT ALADIN

We now present the main result of this paper. Specifically, in this section we address two major challenges of ALADIN in resource allocation problems: (i) the high communication load caused by transmitting Hessian and Jacobian information, and (ii) the computational cost of solving large-scale coupled QPs. To address these issues, we integrate an adjoint-based BFGS scheme that allows the coordinator to reconstruct accurate Hessian and Jacobian approximations without explicit transmission. Furthermore, we introduce an event-triggered update scheme, which keeps local matrices fixed between triggers. This approach significantly improves computational efficiency while preserving convergence.

4.1 Efficient Adjoint BFGS ALADIN

Although the above algorithm can solve the distributed optimization problem (1) with high accuracy, it requires substantial uplink and downlink communication. This section proposes an adjoint-based BFGS scheme to address this issue. The proposed algorithm, referred to as *Efficient Adjoint BFGS ALADIN*, is summarized in Algorithm 1.

Following the algorithmic framework in (3), the proposed method consists of four steps: (i) Following the first step of (3), each node solves its local NLP subproblem to compute the updated primal variable $x_i^{[k+1]}$ and the associated multiplier $\mu_i^{[k+1]}$. (ii) Based on $(x_i^{[k+1]}, \mu_i^{[k+1]})$, it computes the local gradient $v_i^{[k+1]}$ and updates auxiliary vectors $(S_i, d_i, z_i, \sigma_i, \gamma_i^{[k+1]})$ as defined in (7) and (8). The updated quantities $(S_i, d_i, z_i, \sigma_i, \gamma_i^{[k+1]})$ are then transmitted to the coordinator¹. (iii) Using the adjoint-based BFGS update described in Diehl et al. (2010) and shown in (9) and (10), the coordinator refines the Hessian and Jacobian approximations with high accuracy. (iv) Finally, the dual variables $\lambda^{[k+1]}$ and $\tilde{\mu}_i$ together with the primal direction Δx_i are updated according to (11) and (12) for all $i \in \{1, 2, \dots, N\}$.

Remark 1. Reduction in Communication Overhead. The primary distinction from (3) lies in the third step, where the proposed algorithm avoids both explicit construction and transmission of full sensitivity matrices

¹ Transmitting either variable differences or the variables themselves incurs the same communication load. This work adopts the former approach, where variable differences are computed in parallel across all nodes.

Algorithm 1 Efficient Adjoint BFGS ALADIN

Initialization: Initial guess of dual variable λ and primal variables $\{Y_i\}$, $\forall i$, choose $\rho > 0$.

Output: Optimal solution $\{Y_i^*\}$.

Repeat:

1. Paralleled solve local NLP:

$$(x_i^{[k+1]}, \mu_i^{[k+1]}) = \arg \min_{x_i} f_i(x_i) + (\lambda^{[k]})^\top A_i x_i + \frac{\rho}{2} \|x_i - y_i^{[k]}\|^2$$

$$\text{s.t. } g_i(x_i) = 0 \mid \mu_i. \quad (6)$$

2. Evaluate the local gradient $v_i^{[k+1]} = \nabla f_i(x_i^{[k+1]})$, and update the auxiliary quantities required for the Hessian and Jacobian recovering:

$$\begin{cases} S_i = x_i^{[k+1]} - x_i^{[k]}, & d_i = v_i^{[k+1]} - v_i^{[k]}, \\ z_i = g_i(x_i^{[k+1]}) - g_i(x_i^{[k]}), & \sigma_i = \mu_i^{[k+1]} - \mu_i^{[k]}. \end{cases} \quad (7)$$

Then compute the adjoint direction using the reverse mode

$$\gamma_i^{[k+1]} = \nabla_{x_i} g_i(x_i^{[k+1]})^\top \sigma_i^{[k+1]}. \quad (8)$$

3. Update the BFGS Hessian approximation as

$$H_i^{[k+1]} = H_i^{[k]} - \frac{H_i^{[k]} S_i S_i^\top H_i^{[k]}}{S_i^\top H_i^{[k]} S_i} + \frac{d_i d_i^\top}{S_i^\top d_i}. \quad (9)$$

Moreover, the Jacobian approximation updates as showed in Diehl et al. (2010)

$$C_i^{[k+1]} = C_i^{[k]} + \frac{(z_i - C_i^{[k]} S_i) \left((\gamma_i^{[k+1]})^\top - \sigma_i^\top C_i^{[k]} \right)}{\left((\gamma_i^{[k+1]})^\top - \sigma_i^\top C_i^{[k]} \right) S_i}. \quad (10)$$

4. Update the global dual variable λ with the parameters in (4):

$$\lambda^{[k+1]} = \left(\sum_{i=1}^N G_i - Q_i R_i^{-1} Q_i^\top \right)^{-1} p. \quad (11)$$

Paralleled update local primal and dual variables:

$$\begin{cases} \tilde{\mu}_i = -R_i^{-1} (C_i^{[k+1]} H_i^{-1} v_i + Q_i^\top \lambda^{[k+1]}), \\ y_i^{[k+1]} = x_i^{[k+1]} - H_i^{-1} (v_i + (C_i^{[k+1]})^\top \tilde{\mu}_i + A_i^\top \lambda^{[k+1]}). \end{cases} \quad (12)$$

(Hessian and Jacobian), thereby reducing computational and communication overhead. In (3), each node transmits the tuple $(x_i^{[k+1]}, v_i^{[k+1]}, H_i, C_i)$ to the coordinator, yielding a communication load of $(2n_i + n_i^2 + c_i \times n_i)$. For least-squares problems under the *Gauss-Newton ALADIN* scheme, the Hessian transmission is replaced by that of the measurement Jacobian, as discussed in Du et al. (2019). In contrast, Algorithm 1 reconstructs accurate Hessian and Jacobian approximations at the coordinator using the adjoint-based BFGS update. As a result, each node only transmits $(S_i, d_i, z_i, \sigma_i, \gamma_i^{[k+1]})$, as defined in (7) and (8), reducing the communication load to $(2n_i + 3c_i)$.

As shown in (4), the coordinator update involves repeated matrix inversions, which remain the main computational bottleneck. This issue is alleviated in the following subsection.

4.2 Real-Time Efficient Adjoint BFGS ALADIN

This subsection introduces an event-triggered update scheme to accelerate Algorithm 1. Inspired by (Houska and Shi, 2022, Algorithm 1, Step 9), the Hessian and Jacobian approximations $H_i^{[k]}$ and $C_i^{[k]}$, together with the associated

matrices G_i , Q_i , and R_i defined in (4), are updated only when $\log_3(k) \in \mathbb{N}$ and kept constant otherwise. This design is motivated by the observation that, as the iterates $x_i^{[k+1]}$ and $x_i^{[k]}$ gradually converge, the sensitivity matrices gradually stabilize, making frequent updates unnecessary. Between two triggering iterations, only the local NLP solves in (6) and the coordination updates in (11)–(12) are performed, while the sensitivity matrices in (9)–(10) are reused. Hence, the proposed heuristic reduces the update frequency over time without compromising convergence performance. Unlike Houska and Shi (2022), we extend this event-triggered sensitivity strategy to the nonconvex setting considered in this paper. Moreover, by exploiting the closed-form solution of the coordination QP in (11)–(12), the proposed approach eliminates the need for external QP solvers. Applying this strategy to both the Efficient Gauss-Newton ALADIN (as shown in Section 3) and the proposed Efficient Adjoint BFGS ALADIN (Algorithm 1) yields their real-time variants: *RT-Efficient Gauss-Newton ALADIN* and *RT-Efficient Adjoint BFGS ALADIN*.

4.3 Local Convergence Analysis

The convergence analysis of Algorithm 1 follows the same line as in (Houska et al., 2016, Section 7); therefore, only a proof sketch is provided here due to space constraints. From (Du et al., 2019, Theorem 1), let the initial iterate of Algorithm 1 be sufficiently close to the optimal solution x^* . Then, under Assumptions 1 and 2 and provided that the local NLPs (6) are solved exactly, there exists a constant $\kappa < \infty$ such that

$$\left\| \begin{matrix} x_i^{[k+1]} - y_i^* \\ \mu_i^{[k+1]} - \mu_i^* \end{matrix} \right\| \stackrel{(6)}{\leq} \kappa \left\| \begin{matrix} y_i^{[k]} - y_i^* \\ \lambda^{[k]} - \lambda^* \end{matrix} \right\|. \quad (13)$$

Furthermore, using the efficient adjoint BFGS information generated by (7)–(12), the following inequality holds:

$$\left\| \begin{matrix} y^{k+1} - y^* \\ \lambda^{[k+1]} - \lambda^* \end{matrix} \right\| \leq \left\| \begin{matrix} y^{[k+1]} - y^* \\ \lambda^{[k+1]} - \lambda^* \end{matrix} \right\| \stackrel{(7)-(12)}{\leq} \gamma \left\| \begin{matrix} x^{[k+1]} - y^* \\ \lambda^{[k]} - \lambda^* \end{matrix} \right\|. \quad (14)$$

Here γ quantifies the discrepancy between the Hessian approximation generated by (9) and the exact Hessian; this error can be made sufficiently small, as guaranteed by (Diehl et al. (2010)). Combining (13) and (14), we obtain that if $\gamma(\kappa + 1) < 1$, the following contraction holds:

$$\left\| \begin{matrix} y^{[k+1]} - y^* \\ \lambda^{[k+1]} - \lambda^* \end{matrix} \right\| \stackrel{(13),(14)}{\leq} \gamma(\kappa + 1) \left\| \begin{matrix} y^{[k]} - y^* \\ \lambda^{[k]} - \lambda^* \end{matrix} \right\|. \quad (15)$$

This establishes the local convergence of Algorithm 1. Note that the above analysis also applies to the algorithm discussed in Section 4.2.

5. NUMERICAL EXPERIMENT

This section evaluates the proposed *Efficient Adjoint-BFGS ALADIN* and *Efficient Gauss-Newton ALADIN* Wu et al. (2025) methods on the differential drive robots problem (which is a practical nonconvex MHE problem) and also presents their real-time implementations.

5.1 Time-splitting Based MHE Model

The corresponding system dynamics and observation model are defined by

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

Here, the vector $x = (\phi, \psi, \theta)^\top$ involves three state variables, representing the lateral position ϕ , longitudinal position ψ , and orientation angle θ . The observation vector $y = (r, \alpha)^\top$ consists of the relative range r and bearing α , while the control inputs $u = (v, \omega)^\top$ include the linear and angular velocities, v and ω , respectively. The measurement noise terms ν_r and ν_α follow Gaussian distributions $\nu_r \sim \mathcal{N}(0, \sigma_r^2)$ and $\nu_\alpha \sim \mathcal{N}(0, \sigma_\alpha^2)$.

Based on the above control system, the resulting MHE problem at each time step l is formulated by penalizing the system distribution and the fitting error over a prediction horizon of length L , and solved in a receding-horizon manner. The corresponding simplified objective function can be expressed as follows:

$$\min_x \frac{1}{2} \|x_{l-L} - \hat{x}_{l-L}\|_{P^{-1}}^2 + \frac{1}{2} \sum_{n=l-L}^l \|h(x_n) - y_n\|_{V^{-1}}^2 \quad (16)$$

where $\hat{x}_{l-L}$ represents the prior state estimate, $P \in \mathbb{R}^{|x_n| \times |x_n|}$ and $V \in \mathbb{R}^{|y_n| \times |y_n|}$ are the covariance matrix of the initial state estimation error and measurement noise, respectively, and $h: \mathbb{R}^{|x_n|} \rightarrow \mathbb{R}^{|y_n|}$ denotes the nonlinear measurement function.

To apply the proposed algorithm, we adopt the time-splitting strategy from Wu et al. (2025)), which partitions the prediction horizon into N consecutive sub-windows. For each sub-window indexed by i , auxiliary boundary variables $z_i = ((z_i^a)^\top, (z_i^b)^\top)^\top$ are introduced to represent the initial and terminal states, respectively. Specifically, z_i^a denotes the initial state $z_i^a = x_{l-L+(i-1)t}$, while a new auxiliary variable z_i^b corresponds to the terminal state coupled with the next sub-window via the constraint $z_i^b = z_{i+1}^a$. Collectively, the optimization variables of the i -th sub-problem are defined as $X_i = ((z_i^a)^\top, (\tilde{x}_{(i)})^\top, (z_i^b)^\top)^\top$, $X_i \in \mathbb{R}^{|X_i|}$, and the corresponding local objective is given as, for $i = 1$ and $i = N$,

$$\begin{cases} J_1(X_1) = \frac{1}{2} \|z_1^a - \hat{x}_{l-L}\|_{P^{-1}}^2 + \frac{1}{2} \sum_{j=l-L}^{l-L+t-1} \|h(x_j) - y_j\|_{V^{-1}}^2, \\ J_N(X_N) = \frac{1}{2} \sum_{j=l-L+(N-1)t}^l \|h(x_j) - y_j\|_{V^{-1}}^2, \end{cases} \quad (17)$$

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

$$J_i(X_i) = \frac{1}{2} \sum_{j=l-L+(i-1)t}^{l-L+it-1} \|h(x_j) - y_j\|_{V^{-1}}^2. \quad (18)$$

Similarly, the nonlinear dynamic equality constraints are partitioned into independent sub-vectors according to the same time-splitting scheme, for $i = 1, \dots, N-1$,

$$\mathcal{F}_i(X_i) = \begin{bmatrix} x_{l-L+(i-1)t+1} - f(z_i^a, u_{l-L+(i-1)t}) \\ x_{l-L+(i-1)t+2} - f(x_{l-L+(i-1)t+1}, u_{l-L+(i-1)t+1}) \\ \vdots \\ z_i^b - f(x_{l-L+it-1}, u_{l-L+it-1}) \end{bmatrix}, \quad (19)$$

for $i = N$:

$$\mathcal{F}_i(X_i) = \begin{bmatrix} x_{l-L+(i-1)t+1} - f(z_i^a, u_{l-L+(i-1)t}) \\ x_{l-L+(i-1)t+2} - f(x_{l-L+(i-1)t+1}, u_{l-L+(i-1)t+1}) \\ \vdots \\ x_l - f(x_{l-1}, u_{l-1}) \end{bmatrix}. \quad (20)$$

Consequently, based on (17)-(20), the MHE problem formulated under the time-splitting scheme can be represented as a general distributed optimization problem, as shown in (1), where the consistency constraints $z_i^b = z_{i+1}^a$ are equivalently expressed as affine coupling constraints in (1). The explicit form of A_i can be found in Wu et al. (2025)).

The prediction horizon is set to 25, divided into 4 parallel sub-windows. The initial position and orientation of the robot are initialized as $x_0 = (0.1, 0.1, 0.0)^\top$, and the reference trajectory $x^* = (\phi^*, \psi^*, \theta^*)^\top$ is generated by MPC under the same control model. The MHE decision variables are initialized at $(\phi^*, \psi^*, 0)^\top$. In the proposed *Efficient Adjoint BFGS ALADIN* setup, the penalty parameter is set to $\rho = 25$, and the dual variables λ and μ are initialized to zero. At the first iteration, each node sets $H_i = I$ and initializes C_i with the exact Jacobian. All simulations are executed on a desktop equipped with a 2.5 GHz Intel i5-13490F processor (10 cores, 16 threads) and 16 GB of RAM, using Casadi-3.6.6 with IPOPT in MATLAB R2024b on Windows 11.

5.2 Numerical Results

Fig. 1 compares the convergence behavior of the proposed *Efficient Adjoint BFGS ALADIN*, *Efficient Gauss-Newton ALADIN*, and their real-time variants for solving the MHE problem with $N = 4$. Both real-time implementations adopt a $\log_3(k) \in \mathbb{N}$ -based sensitivity update strategy, corresponding to three updates within 50 iterations. All four methods exhibit linear convergence during sensitivity updates and achieve high-accuracy solutions. *Efficient Adjoint-BFGS* attains 10^{-9} accuracy in 15 iterations, while *Efficient Gauss-Newton ALADIN* reaches 10^{-11} in 25 iterations. Their real-time counterparts, with fewer sensitivity updates, converge in a two-stage pattern to 10^{-9} and 10^{-7} within approximately 35 iterations.

Table 1 provides the total CPU time of the two efficient ALADIN variants and their real-time counterparts under different sub-window partitions. Since the sub-problem solutions in ALADIN rely on external solvers, their runtimes are nondeterministic and negligible for large-scale cases. Hence, only the CPU time for QP construction and solution in the coordination step is reported. The results show that real-time implementations significantly reduce computation time. Among all methods, *Efficient Gauss-Newton ALADIN* exhibits the longest runtime, while its real-time counterpart achieves the shortest. Unlike the asymptotic relation $N^* \approx \sqrt{L}$ reported in (Wu et al. (2025)), the optimal sub-window partition here deviates from the trend because the real-time variants skip sensitivity updates and matrix inversions at each iteration.

6. CONCLUSIONS

This paper proposed a novel distributed algorithm, termed *Real-Time Efficient Adjoint BFGS ALADIN*, for resource

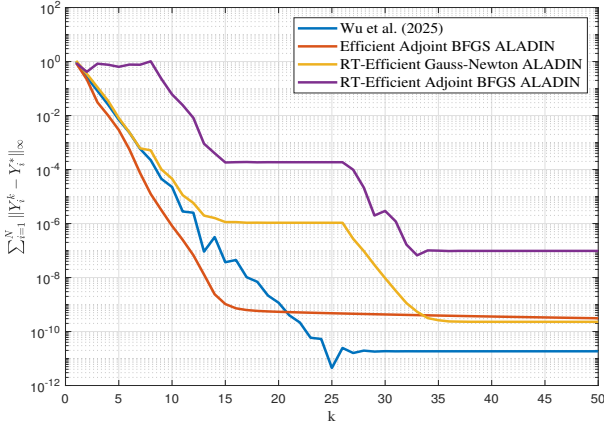


Fig. 1. Convergence comparison of Efficient Adjoint BFGS ALADIN (Algorithm 1) with Efficient Gauss-Newton ALADIN (in Wu et al. (2025)), RT-Efficient Gauss-Newton ALADIN and RT-Efficient Adjoint BFGS ALADIN (in Section 4.2).

N	Efficient Gauss-Newton ALADIN	Efficient Adjoint BFGS ALADIN	RT-Efficient Gauss-Newton ALADIN	RT-Efficient Adjoint BFGS ALADIN
3	3.000	1.40	0.268	0.486
4	2.300	0.971	0.232	0.516
5	1.400	0.973	0.205	0.633
6	1.500	0.821	0.199	0.727

Table 1. Total CPU time [ms] for different algorithms across N sub-windows (measured over 50 iterations).

allocation problems. The method efficiently handles non-convex optimization while ensuring linear-scaling communication overhead. Furthermore, we presented an event-triggered mechanism that avoids updating local Hessian and Jacobian matrices at every iteration, thereby reducing computational burden. Numerical simulations validated the algorithm's advantages over existing approaches.

REFERENCES

- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1), 1–122.
- Diehl, M., Walther, A., Bock, H.G., and Kostina, E. (2010). An adjoint-based SQP algorithm with quasi-Newton Jacobian updates for inequality constrained optimization. *Optimization Methods & Software*, 25(4), 531–552.
- Doostmohammadian, M., Aghasi, A., Pirani, M., Nekouei, E., Zarrabi, H., Keypour, R., Rikos, A.I., and Johansson, K.H. (2025). Survey of distributed algorithms for resource allocation over multi-agent systems. *Annual Reviews in Control*, 59, 100983.
- Du, X., Engelmann, A., Jiang, Y., Faulwasser, T., and Houska, B. (2019). Distributed state estimation for AC power systems using Gauss-Newton ALADIN. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 1919–1924. IEEE.
- Du, X., Johansson, K.H., and Rikos, A.I. (2025). Decentralized optimization via RC-ALADIN with efficient quantized communication. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, 4357–4363. IEEE.
- Du, X. and Wang, J. (2023). Consensus ALADIN: A framework for distributed optimization and its application in federated learning. *arXiv preprint arXiv:2306.05662*.
- Du, X. and Wang, J. (2025). Distributed consensus optimization with consensus ALADIN. In *2025 American Control Conference (ACC)*, 2949–2955.
- Engelmann, A., Jiang, Y., Mühlfordt, T., Houska, B., and Faulwasser, T. (2019). Towards distributed OPF using ALADIN. *IEEE Transactions on Power Systems*, 34(1), 584–594.
- Fedorová, K. (2025). *Structure-Exploiting Model Predictive Control for Real-Time and Embedded Applications*. Ph.D. thesis, Slovak University of Technology in Bratislava, Bratislava.
- Feng, X., Di Cairano, S., and Quirynen, R. (2020). Inexact adjoint-based SQP algorithm for real-time stochastic nonlinear MPC. *IFAC-PapersOnLine*, 53(2), 6529–6535.
- Hespanhol, P. and Quirynen, R. (2021). Adjoint-based SQP method with block-wise quasi-Newton Jacobian updates for nonlinear optimal control. *Optimization Methods and Software*, 36(5), 1030–1058.
- Hong, M., Luo, Z.Q., and 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.
- Houska, B., Frasch, J., and Diehl, M. (2016). An augmented lagrangian based algorithm for distributed non-convex optimization. *SIAM Journal on Optimization*, 26(2), 1101–1127.
- Houska, B. and Shi, J. (2022). Distributed MPC with ALADIN-A tutorial. In *2022 American Control Conference (ACC)*, 358–363.
- Kouzoupis, D., Quirynen, R., Houska, B., and Diehl, M. (2016). A block based ALADIN scheme for highly parallelizable direct optimal control. In *2016 American Control Conference (ACC)*, 1124–1129.
- Lanza, L., Faulwasser, T., and Worthmann, K. (2025). Distributed optimization for energy grids: A tutorial on ADMM and ALADIN.
- Quirynen, R., Gros, S., and Diehl, M. (2018). Inexact Newton-type optimization with iterated sensitivities. *SIAM Journal on Optimization*, 28(1), 74–95.
- Ren, X., Bastianello, N., Johansson, K.H., and Parisini, T. (2026). Communication-efficient stochastic distributed learning. *IEEE Transactions on Automatic Control*.
- Stomberg, G., Engelmann, A., Diehl, M., and Faulwasser, T. (2025). Decentralized real-time iterations for distributed NMPC. *IEEE Transactions on Automatic Control*.
- Wang, Y., Wu, S., Yang, G., Chu, J., Rikos, A.I., and Du, X. (2025). ALADIN- β : A distributed optimization algorithm for solving MPCC problems. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, 5329–5334. IEEE.
- Wu, S., Wang, Y., Wang, J., Rikos, A.I., and Du, X. (2025). A time splitting based optimization method for nonlinear MHE. In *2025 IEEE 64th Conference on Decision and Control (CDC)*, 5335–5341. IEEE.