Energy and AI 21 (2025) 100536

Contents lists available at ScienceDirect

Energy and AI

journal homepage: www.elsevier.com/locate/egyai

Distributed nonlinear model predictive control for building energy systems: An ALADIN implementation study

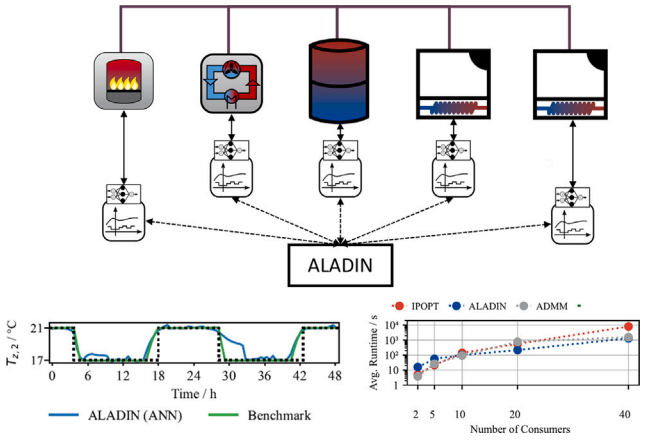
Steffen Eser *, Ben Spoek, Augustinus Schütz, Phillip Stoffel, Dirk Müller

RWTH Aachen University, E.ON Energy Research Center, Institute for Energy Efficient Buildings and Indoor Climate, Mathieustr. 10, Aachen, 52074, Germany

HIGHLIGHTS

- Nonlinear model predictive control for buildings with multivariate coupling.
- ALADIN comes close to performance of central MPC while ADMM fails.
- ALADIN is less sensitive to parameter and scaling variations than ADMM.
- Neural Network based models reduce computation time compared to ODE-based approaches.
- DMPC with ALADIN is faster than central MPC starting at 10 thermal zones.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Multi agent system
HVAC
Control
Distributed model predictive control
ALADIN
ADMM
Data driven model predictive control
Model predictive control

ABSTRACT

The implementation of sophisticated control strategies for building energy systems is crucial for improving energy efficiency and occupant comfort. While nonlinear model predictive control offers promising benefits, its application to large-scale building systems remains challenging due to computational complexity and system coupling. This work presents a comprehensive study of Nonlinear Distributed Model Predictive Control (NDMPC) implementation for building energy systems, comparing Alternating Direction Method of Multipliers (ADMM) and Augmented Lagrangian Alternating Direction Inexact Newton (ALADIN) algorithms alongside different modeling approaches. We examine a multi-zone heating system with thermal storage and multiple producers, investigating both Ordinary Differential Equation (ODE)-based and Artificial Neural Network (ANN) based modeling strategies. Through systematic parameter tuning using Bayesian optimization and closed-loop scaling analysis with up to 40 thermal zones, we demonstrate that ALADIN-based NDMPC can achieve performance comparable to centralized model predictive control, showing greater robustness to parameter variations than ADMM. Our results reveal that ANN-based models effectively mitigate distributed integration errors and significantly reduce computation time compared to ODE-based approaches. Detailed computational profiling identifies specific bottlenecks in different NDMPC components. These findings advance the practical implementation of NDMPC in building energy systems, offering concrete strategies for modeling choices, parameter tuning, and system architecture design.

* Corresponding author.
E-mail address: steffen.eser@eonerc.rwth-aachen.de (S. Eser).
URL: https://www.ebc.eonerc.rwth-aachen.de/cms/~dmzz/E-ON-ERC-EBC/ (S. Eser).

https://doi.org/10.1016/j.egyai.2025.100536
Received 12 March 2025; Received in revised form 3 June 2025; Accepted 5 June 2025
Available online 19 June 2025

1. Introduction

The modernization of building control systems is crucial for reducing energy consumption and CO_2 emissions in the built environment. MPC has emerged as a promising approach, offering several advantages over traditional building controllers [1]. These traditional controllers typically rely on rule-based systems and simple Proportional-Integral-Derivative (PID) or hysteresis controllers for setpoint tracking. In contrast, MPC can implement precise heating and cooling schedules, optimize component operations, and consider factors such as weather forecasts and dynamic energy prices [2]. A recent critical review of MPC field implementations by Saloux et al. [3] underscores its practical success, demonstrating average energy cost reductions of approximately 25% across numerous studies. However, the studies reviewed by Saloux et al. [3] often featured buildings with a floor area of less than 1000 m^2. Realizing the full potential of MPC in large-scale, complex BESs presents ongoing challenges. To address the challenges of applying MPC to large-scale and complex BESs, DMPC has emerged as a key research area [4,5]. DMPC offers the potential for improved scalability by decomposing the overall control problem into smaller, coupled subproblems that can be solved in parallel by individual agents. This modularity can also simplify modeling and system maintenance [6]. However, BESs are inherently nonlinear systems, involving thermal dynamics, hydraulic interactions, and varying component efficiencies. Therefore, NDMPC is often necessary to accurately capture these characteristics and achieve optimal performance.

1.1. Literature review

This section provides an overview of the current state of research in DMPC for BES. We first we discuss some case studies on DMPC for buildings, then we discuss solution algorithms and challenges specific to nonlinear DMPC.

1.1.1. Current state of distributed model predictive control for buildings

Applications of DMPC in buildings largely fall into two categories: smart grids and multi-zone control.

In the context of smart grids, Shi et al. [7] applied DMPC to control a smart grid with linear building models coupled through AC power flow. Their approach used custom routines within an ADMM-based distributed optimization framework to handle non-convex constraints resulting from the AC power flow equations. For a demonstration on the IEEE-123-bus system partitioned into 3 agents, their DMPC performed within 1% of the centralized MPC. While computing time approximately doubles compared to centralized computation, the advantage lies in reduced information exchange.

Lefebure et al. [8] explored DMPC based on dual decomposition for multi-building energy hubs. They employed a mixed-integer quadratic programming approach centrally, relaxing it to reduce the number of integer decisions in the distributed version. They used regularization terms and augmented objective functions to improve the convergence of their dual ascent approach. They achieved closed-loop performance equivalent to the centralized controller, scaling better than the centralized controller with larger networks.

For multi-zone control, Lin and Adetola [9] utilized ADMM-based NDMPC to control a simulated multi-zone building with up to 320 zones, conditioned with a central AHU with variable air volume flow per zone. They included a cubic objective for the total ventilator power intake and a simple, yet nonlinear modeling for the air temperature per zone. The zones are coupled through the sum of their air mass flows, and a unified supply air temperature. Crucially, they exclude thermal energy consumption from their objective, removing the interdependence between their coupling variables. Their approach demonstrated linear scaling with the number of zones.

Yang et al. [10] examine a NDMPC problem where they control the ventilation rate for a multi-zone building cooled by an air handling unit. The MPC has a nonlinear objective including cooling power and fan power, linear thermal coupling between zones and bi-linear zone dynamics including the ventilation rate and the zone temperature. They use auxiliary variables and McCormick envelopes for the bilinear constraints to relax the zone agents into Quadratic Programs (QPs), while retaining the nonlinear objective in the fan agent. They solve the distributed Optimal Control Problem (OCP) using the accelerated distributed augmented Lagrangians (ADAL) method, demonstrating good scalability for systems up to 500 zones.

While these studies solved challenging DMPC problems, they leave some aspects of NDMPC unexplored. In particular, they do not feature multi-variate nonlinear coupling of system dynamics between subsystems. These kinds of couplings easily arise in hydraulic systems where components are connected with a water mass flow, a supply and a return temperature. While the study in [9] appears to make such a coupling for an air-based system, their simplifications with regard to the producer side make the multi-variate coupling inconsequential.

1.1.2. Solution algorithms for distributed model predictive control

Early works on DMPC focused on the distinction between cooperative and non-cooperative schemes. The book by Maestre and Negenborn [5] comprises 35 contributions on DMPC featuring different authors. Non-cooperative DMPC often involved non-iterative methods, while cooperative DMPC required global system knowledge across all agents [5]. The seminal work by Rawlings et al. [11] on MPC fundamentals covers linear DMPC schemes that necessitate the exchange of system matrices. A comprehensive review by Christofides et al. [12] in 2013 also focuses on the cooperative schemes, with only brief mentions of distributed optimization based methods like dual decomposition, or augmented Lagrangian techniques.

Recent years have witnessed a paradigm shift towards distributed optimization-based approaches in optimal control for buildings and grids. These methods offer individual agents considerable autonomy, requiring only the exchange of coupling vectors while effectively managing strong inter-system couplings. Primal–dual and augmented Lagrangian-based approaches have emerged as popular choices, being featured in all references provided in the above Section 1.1.1.

Among these, ADMM has gained significant traction due to its simplicity, versatility and effectiveness. ADMM was originally proposed in the 70s [13,14] and renewed by Boyd et al. [15]. A comprehensive survey by Yang et al. [16] provides an in-depth analysis of ADMM and its many variants, highlighting its wide-ranging applications in distributed optimization problems. However, when applied to non-convex problems, ADMM's performance can be suboptimal, often resulting in slow convergence. In fact, Engelmann [17] notes that for constrained optimization problems, both convergence speed and solution accuracy can be limited, regardless of parameterization. To address the limitations of ADMM in non-convex scenarios, more advanced algorithms have been developed. Sun and Sun [18] propose a two-level algorithm to handle distributed non-convex optimization problems. They introduce slack variables to ensure feasibility and convergence of the inner ADMM level, while driving feasibility at the outer level, using an augmented Lagrangian based approach. Tang and Daoutidis [19] expand on the two level algorithm, giving suggestions for parameterization and realizing speedups through approximate iterates and Anderson Acceleration. They demonstrate the algorithm on a quadruple tank process, controlling volume flows through NDMPC.

Another algorithm for non-convex distributed optimization is ALADIN, introduced by Houska et al.. ALADIN has shown promising results in handling non-convex problems more effectively, combining the decomposition benefits of ADMM with the fast local convergence of Newton-type methods. Compared with ADMM, ALADIN trades a better convergence rate for increased communication and central computation overhead. While wide-spread adoption of ALADIN has yet to manifest, there already exist a number of applications and improvements on the original algorithm. Jiang et al. [21] used ALADIN for a DMPC

in smart grids. They used relaxed decoupled equality constraints in the consensus QP and dynamic updates of the slackness parameters for said relaxation, improving upon the ADMM based solution both in closed loop number of iterations, and in scaling towards larger systems [21]. Su et al. [22] used ALADIN to coordinate demand response in commercial buildings. While their problem formulation was convex, they used a modified ALADIN variant with an analytical central step to speed up convergence compared to ADMM [22]. Engelmann et al. [23] extended the standard ALADIN algorithm by solving the coordination QP in a distributed fashion as well, allowing for a fully distributed implementation [17,23].

1.1.3. Issues with distributed transcription of optimal control problems

While existing research on NDMPC often focuses on specific optimization algorithms, the distributed transcription is rarely a focus. Direct transcription methods, such as direct multiple shooting and direct collocation are popular for central MPC, as they are easy to implement and scale well for larger problems [24]. Multiple shooting requires a discrete mapping from the current system states onto a future time step, commonly realized through integrating ODEs over the time step using Runge–Kutta methods or robust solvers like CVODE for stiff systems. Collocation methods approximate the state evolution over a time interval through polynomials, simultaneously optimizing the polynomial roots with the other decision variables.

The application of direct nonlinear transcription methods towards distributed optimization faces a challenge. As the neighbor-to-neighbor communication between agents consists of vectors of the shared variables' values at discrete points in time, the continuous evolution of neighboring states is not accurately considered in an agents' integration scheme, resulting in a reduced integration order with regard to some states. A recent study on real-time interactions for NDMPC [25] acknowledged this problem, referring to a distributed multiple shooting method [26]. While the distributed transcription error can be mitigated, i.e. by using the proposed distributed multiple shooting method, or using a finer discretization grid, these techniques come with the tradeoff of increasingly large coupling variables.

It is worth noting that many applications driving the development of distributed algorithms for non-convex optimization, such as optimal power flow problems, often assume separable time scales and ignore transient effects [17]. While this simplification sidesteps the issue of reduced integration order, it may not be applicable in dynamic building energy systems where transient effects play a significant role.

1.1.4. Modeling for nonlinear model predictive control

In the literature, the categorization of models into white-box, gray-box and black box is overwhelmingly prevalent, i.e. in studies [2,27,28]. Yet, the boundaries between these categories are often blurred, making it difficult to define precisely when a model transitions from one category to another. This work adopts a different perspective, shifting the focus from the *source* of the model knowledge (data vs. expert) to the *structure* of the model itself (continuous vs. discrete).

Continuous models, typically based on ODEs, can represent the underlying physics of the system, potentially leading to better generalization. There are many existing studies using these models for Nonlinear Model Predictive Control (NMPC), for example in studies [28–32]. However, deriving and parameterizing ODE-based models can be complex, particularly for large-scale BES with intricate dynamics. Furthermore, as discussed in Section 1.1.3, using continuous models within a distributed optimization framework introduces challenges related to numerical integration.

Discrete, data-driven models offer an alternative approach. These models learn nonlinear relationships directly from data, bypassing the need for explicit physical equations. Nonlinear Autoregressive with Exogenous Inputs (NARX) models are a powerful class of discrete models capable of capturing temporal dependencies in BES data [33]. Stoffel compare the use of different NARX and Autoregressive with Exogenous Inputs (ARX) models based on ANN and Gaussian Process Regression (GPR) to model thermal zone behavior and heat pump power consumptions for building MPC, showing accurate performance in simulation and real experiments. As these models are inherently discrete, the performance degradation due to neighboring states with unknown dynamics is already priced in, and should not further deteriorate when moving to distributed formulation. In this work, NARX models are implemented using ANNs, leveraging their ability to approximate arbitrary nonlinear functions.

The preceding literature review highlights that while DMPC offers a path towards managing large-scale BESs, significant challenges remain, particularly when addressing the inherent nonlinearities and complex interdependencies prevalent in such systems. Key identified gaps include the need for NDMPC strategies capable of handling multivariate couplings, comprehensive comparisons of advanced distributed optimization algorithms like ALADIN and ADMM in these demanding contexts, and effective methods for mitigating distributed integration errors, especially when using ODE-based models. Successfully addressing these NDMPC challenges is crucial for unlocking more ambitious control objectives. These include the simultaneous optimization of multiple interacting subsystems considering, for example, nonlinear component efficiencies and hydraulic effects, as well as enabling seamless integration with smart grids and the provision of demand response services. This paper, therefore, focuses on the implementation and comparative analysis of NDMPC strategies for a BES characterized by strong hydraulic and nonlinear couplings between producers, thermal storage, and multiple consumer zones. The aim of this work is to rigorously investigate and compare the performance of ALADIN and ADMM for this class of problems. We evaluate the efficacy of ODE-based versus ANN-based modeling approaches within the distributed control framework, with a particular focus on solution quality, computational efficiency, scalability, and strategies for managing the challenges of distributed system representation and optimization.

1.2. Contribution

This work addresses several key challenges in implementing NDMPC for complex BESs. Our main contributions are:

1. Comprehensive NDMPC implementation for BES: We demonstrate a fully-functional NDMPC which couples producers, distribution systems, and consumers in a building energy system. This implementation considers nonlinear dynamics and multivariate couplings, addressing a gap in current literature where such comprehensive NDMPC for BES is understudied.
2. Modeling Approach for Distributed Control:
   (a) We introduce the use of NARX models, implemented with ANNs, comparing them to traditional ODE-based modeling in NDMPC.
   (b) We demonstrate that this ANN-based approach mitigates errors arising from distributed integration without resorting to complex distributed integration schemes.
3. Algorithmic Performance Analysis: We provide a comparative analysis of ADMM and ALADIN for the NDMPC problem, comparing them to the solution provided by the state of the art Nonlinear Program (NLP) solver IPOPT.
4. Scaling and Parameter Tuning Insights:
   (a) We investigate the scalability of our NDMPC approach by analyzing execution times when increasing the system size.
   (b) We highlight the importance of problem scaling and provide insights into parameter tuning for distributed optimization algorithms, particularly ADMM and ALADIN.

5. Topology Considerations: We identify potential issues with star-like topologies in building energy systems and discuss their impact on NDMPC performance.

Through these contributions, we aim to advance the practical implementation of NDMPC in building energy systems, addressing key challenges in modeling, optimization, and scalability.

The remainder of this paper is organized as follows: Section 2 introduces the examined problem class and introduces the case study. Section 3 presents the controller architecture, proposing two distinct modeling approaches, based on differential equations and discrete data-driven models respectively. Section 4 briefly covers ADMM and AL-ADIN, the distributed optimization algorithms used in this study. Section 5 introduces the benchmark controllers against which the NDMPC is compared. Section 6 provides results of our case study, including modeling, parameter tuning, controller performance, and scalability. Section 7 interprets the observed results, assesses the viability of NDMPC, considers limitations of the present work, and recommends future research directions.

2. Case study: Strong hydraulic coupling in building energy systems

2.1. Model predictive control of buildings: Expectations and challenges

Many MPC benefits observed with basic linear MPC, such as pre-heating/cooling strategies and activating systems only when necessary, can also be achieved with rules and PID controllers. In contrast, achieving more sophisticated goals, like simultaneous optimization of temperature and mass flow or considering nonlinear efficiency curves, requires nonlinear optimization and consideration of whole-building couplings. For large, interconnected building systems in particular, this leads to intractable MPC formulations, making ambitious goals for MPC seemingly unreachable. We experience a paradox, where easy MPC does not bring the full benefits, while an MPC tailored to optimizing all aspects of the operation is difficult to implement.

To address these challenges and fully realize the potential of building MPC, an MPC scheme that considers nonlinear behavior while scaling gracefully with system size is required. Agent-based MPC, or DMPC allows building an MPC from smaller agents that can be configured independently and are computationally tractable. Through parallel implementations of distributed optimization algorithms, these DMPC schemes promise to scale better with system size compared to classic centralized optimization algorithms, like interior point methods, or sequential quadratic programming. In the following, we consider types of agents that can be extracted from a BES, how they are coupled, and present an example BES that is suitable for examining the behavior of NDMPC.

2.2. Coupling within building energy systems

Building energy systems comprise several interconnected subsystems. We categorize the subsystems into three main types:

Rooms/thermal zones. These are the primary interfaces with building occupants, typically equipped with temperature and CO_2 sensors, along with various actuators. Thermal zones are typically equipped with heat emitters (e.g., radiators, underfloor heating) or cooling emitters (e.g., ceiling panels, fan coil units). Some systems, like Variable Air Volumes (VAVs), can provide heating, cooling, and fresh air supply, acting as conditioned air delivery systems rather than simple emitters. While further subdivision into individual components is theoretically possible, we consider thermal zones as a single subsystem for the scope of this study.

The behavior of these subsystems is quite individual and highly influenced by disturbances (e.g. ambient temperature, irradiative or internal gains), however usually exhibits only weak nonlinearities. In many conventional control approaches, parameters like supply temperatures for heating or cooling circuits are determined a priori (e.g., via a heating curve), with local PID controllers then modulating mass flows through valves or dampers. In contrast, MPC allows for the simultaneous optimization of both supply temperatures and mass flows, which can introduce beneficial, yet challenging, bilinear dynamics into the control problem. The primary objective for these subsystems is occupant comfort, typically achieved by maintaining air temperature and CO_2 levels within acceptable ranges.

Producers. These subsystems are responsible for heating and cooling water or air, consuming electricity or fuel in the process. Common examples include gas boilers, combined heat and power units (CHPs), heat pumps, and chillers. The primary objective for producers is typically to minimize energy consumption or operational costs, while keeping supply temperatures adequate. These objectives may be nonlinear due to efficiency curves.

Distribution/storage systems. This category encompasses components such as buffer storages, hydraulic switches, injection circuits, mixing circuits, and potentially pipe or air duct segments when flow times are significant. These systems may have nonlinear dynamics, depending on the degree of modeling, and the degree of freedom (i.e. are some mass flows or temperatures fixed) They generally lack individual objective functions and serve to connect other subsystems.

The coupling between these subsystems is achieved through shared variables, which we refer to as coupling variables. These variables are subject to two types of constraints:

- **Exchange constraints:** These require that the sum of all instances of a variable across subsystems equals zero. For example, the mass flow exiting a producer (with positive sign) must add up to zero considering the mass flows through all supplied subsystems (with negative sign).
- **Consensus constraints:** These mandate that a variable maintains the same value across all relevant subsystems. For instance, the supply temperature exiting a producer must match the supply temperature for all zones served by that producer.

While many studies focus on only one type of coupling, they can also be combined, as seen in [9]. In this study, we realize all couplings through consensus constraints.

2.3. Investigated building energy system

To illustrate the application of DMPC in building energy systems, we consider a representative system as shown in Fig. 1. The system under study, depicted in Fig. 1 and previously examined in [35], features two primary heat producers: a gas boiler and a Combined Heat and Power (CHP) unit. These producers feed into a sensible heat storage tank, which acts as a thermal buffer and distribution hub. The storage tank supplies hot water to the heating emitters of multiple thermal zones, generally represented by N_z consumers in Fig. 1.

The systems' parameters are inspired by the E.ON Energy Research Center main building in Aachen, Germany (usable area $A_{use} \approx 6300\ \mathrm{m}^2$, gross volume $V_g \approx 31500\ \mathrm{m}^3$). For initial detailed controller analysis and parameter tuning (as presented in Sections 6.1 and 6.3), a baseline configuration with two distinct thermal zones is employed. The system's scalability is subsequently investigated by increasing the number of consumer zones up to 40, as detailed in the results on system scaling (Section 6.4). The design heating requirement for the overall system is approximately 107.3 $\mathrm{kW_{th}}$, calculated for a design point of −5 °C ambient temperature and a 21 °C indoor reference temperature. In the aforementioned two-zone baseline configuration, this load is distributed as 81.3 $\mathrm{kW_{th}}$ for Zone 1 and 26.0 $\mathrm{kW_{th}}$ for Zone 2. The overall thermal capacity of the zones solid mass was estimated based on DIN V 18599-2, which suggests $C_b = A_{use} \cdot 45\ \mathrm{Wh/(m^2\ K)}$

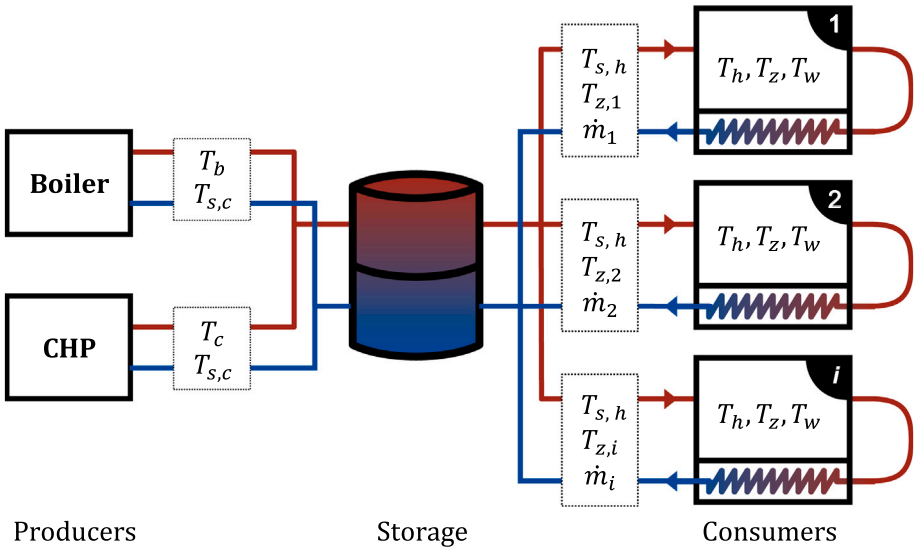


Fig. 1. Controlled system and coupling variables.

for light building zones. Our model utilizes half of this estimated building thermal mass. This change, balanced by the omission of faster heat transfer mechanisms like infiltration and solar gains, ensures appropriately fast system dynamics, which is beneficial for observing control responses within reasonable simulation times. Furthermore, the E.ON ERC building features significant open internal spaces, lending credence to a model with effectively lower overall thermal inertia for its conditioned volume.

The primary focus of this work is the behavior of NDMPC algorithms when faced with strong, nonlinear hydraulic coupling. Therefore, detailed thermal modeling of the zones was considered secondary to the clear representation of inter-agent dependencies. Similarly, the thermal storage is modeled as a stratified tank with two distinct layers. This is a deliberate simplification chosen as a trade-off: it introduces the numerical complexity of enthalpy exchange and distinct temperature levels—key aspects for studying hydraulic coupling in DMPC—without making the problem overly complex for the core algorithmic investigation.

The heat producers are dimensioned as follows. The boiler has a nominal thermal output of 66 kW_{th}, and the CHP unit provides 46 kW_{th}. The CHP is sized smaller than the boiler to encourage its prioritization for base-load coverage, maximizing its operational runtime and overall efficiency. The thermal storage tank has a volume of 15 m^3, providing a thermal capacity of approximately 346 kWh_{th} over a temperature difference of 20 K. This capacity allows the storage to meet the building's total heating requirement for just over three hours. To provide a basis for external thermal disturbances, the simulations utilize an ambient temperature profile from January of the Test Reference Year (TRY) 2015 for Aachen, Germany. However, it is acknowledged that the overall disturbance model is simplified, as solar irradiation and dynamic internal gains are not considered in this study.

For the DMPC, we couple the consumers with the storage through consensus constraints on supply temperature, return temperature, and mass flow simultaneously. This approach results in multiple interacting variables per edge in the system graph, leading to significant computational challenges due to nonlinear interactions. While fixing either mass flow or supply temperature would simplify the problem considerably, it would also limit the potential benefits of NDMPC. In the case of fixed mass flows, the heating of the zones cannot be stopped without draining the storage, while in the case of fixed supply temperatures, the storage cannot provide any flexibility. For the coupling between producers and the storage, we *do* fix the mass flow, balancing the overall challenge of this setup. We allow actuation of relative producer powers and consumer side mass flows, introducing the full complexity of nonlinearly coupled subsystems. For detailed modeling on the physical system, refer to Table 2. Due to different sizes and thermal inertia of the walls, air and heating states, the system requires surprisingly robust integration methods. By analyzing the behavior of the DMPC (and the underlying distributed optimization algorithms), we aim to lay the groundwork for MPC that is able to determine optimal operation of an entire building and not only its subsystems.

3. Design of distributed controller

In this section, we cover the design and implementation of an NDMPC for our example building energy system. We explain how to use neural networks as process models in MPC, state the modeling for our individual problem and derive the optimal control problem for the agents from there.

3.1. Neural networks as state estimators

In the context of NMPC, NARX models based on ANN can be used to form a discrete model of the process, capturing the evolution of a state over a fixed time interval without the need to solve differential equations. We train a single step predictor for the next state based on its past states and external (exogenous) inputs:

$$x(t) = \mathcal{A}(x(t-1), x(t-2), \dots, x(t-n_y), u(t-1), u(t-2), \dots, u(t-n_u)) \quad (1)$$

where $x(t)$ is the system state at time t, $u(t)$ represents the exogenous inputs, and n_y and n_u are the number of past output and input terms used for prediction. Neural networks can be trained to approximate this function $\mathcal{A}$. The network takes as input a window of past states and inputs, and outputs the predicted next state.

Ensuring the stability and robustness of the neural network-based controller remains a nontrivial task. Techniques such as online learning and uncertainty quantification are active areas of research addressing these challenges [34,36,37]. However, the generation and training of these models is not the focus of this work. In this work, we can generate sufficient training data by simulating the building model with a variety of input combinations. Only the most recent past state and input is used.

3.2. Control modeling of subsystems

As illustrated in Fig. 1, the energy system under consideration comprises two producers, a buffer storage, and multiple thermal zones. Table 1 includes a reference to all symbols appearing in the modeling.

Table 1
Summary of symbols used in the BES modeling.

Subsystem	Symbol	Description
General	$\dot{m}$	Mass flow rate
	c_p	Specific heat capacity
	t	Time
	k	Heat transfer coefficient
	C	Thermal capacity
	N_z	Number of zones
Boiler	T_b	Boiler outlet temperature
	u_b	Boiler control input
	$\dot{Q}_{n,b}$	Nominal boiler thermal power
	η_b	Boiler efficiency
	r_{gas}	Gas price
CHP	T_c	CHP outlet temperature
	u_c	CHP control input
	$\dot{Q}_c$	CHP thermal power output
	$P_{el,c}$	CHP electrical power output
	$\dot{Q}_{gas,c}$	CHP gas consumption
	r_{el}	Electricity price
Storage	$T_{s,h}$	Storage hot layer temperature
	$T_{s,c}$	Storage cold layer temperature
	T_{env}	Storage environment temperature
	$\dot{H}_{h\to c}$	Enthalpy exchange between layers
	ϵ	Smoothing parameter
Consumer (Thermal zone)	$T_{h,i}$	Heating system temperature of zone i
	$T_{z,i}$	Zone air temperature
	$T_{w,i}$	Wall temperature
	T_{amb}	Ambient temperature
	$\dot{Q}_{h\to z,i}$	Heat transfer from heating to zone
	$s_{z,i}$	Slack variable for temperature constraint
	w_i	Weight for slack variable in objective
	$T_{z,\min}$	Minimum allowed zone temperature
	$T_{z,\max}$	Maximum allowed zone temperature

Table 2 summarizes the modeling, constraints, and objectives of these subsystems in the MPC framework.

The producers – a CHP and a gas boiler – are modeled using stationary energy balances, assuming negligible thermal inertia. Their respective control inputs, u_c for the CHP and u_b for the boiler, represent normalized power settings ranging from 0 to 1. The boiler's gas usage is inferred from its thermal power output via a constant efficiency, while the CHP's gas consumption and electricity generation are determined through interpolation tables. The producers' objective function incorporates gas usage at price r_{gas} and electricity revenue at price r_{el}.

The thermal storage is modeled as a stratified tank with two distinct layers (hot and cold) to capture basic stratification effects. The two-layer storage model is governed by the differential equations (2) and (3):

$$\frac{C_s}{2}\frac{dT_{s,h}}{dt} = \dot{m}_b c_p T_b + \dot{m}_c c_p T_c - \sum_{i=1}^{N_z} \dot{m}_{h,i} c_p T_{s,h} - \dot{H}_{h\to c} - k_s(T_{s,h} - T_{env}) \quad (2)$$

$$\frac{C_s}{2}\frac{dT_{s,c}}{dt} = \sum_{i=1}^{N_z} \dot{m}_{h,i} c_p T_{h,i} - (\dot{m}_b + \dot{m}_c) c_p T_{s,c} + \dot{H}_{h\to c} - k_s(T_{s,c} - T_{env}) \quad (3)$$

The enthalpy exchange between layers $\dot{H}_{h\to c}$ is defined as:

$$\dot{H}_{h\to c} = \dot{m}_{h\to c} c_p \frac{T_{s,h} + T_{s,c}}{2} + \dot{m}^*_{h\to c} c_p \frac{T_{s,h} - T_{s,c}}{2} \quad (4a)$$

$$\dot{m}_{h\to c} = (\dot{m}_b + \dot{m}_c) - \sum_{i=1}^{N_z} \dot{m}_{h,i} \quad (4b)$$

$$\dot{m}^*_{h\to c} = \sqrt{\dot{m}^2_{h\to c} + \epsilon} \quad (4c)$$

This formulation, utilizing a small smoothing parameter $\epsilon > 0$, ensures the differential equations remain smooth when the net flow between layers reverses, i.e., when the mass flow through producers equals that through consumers. The storage unit is subject only to box constraints on its state variables and has no specific objective function.

Each thermal zone i is characterized by three state variables: heating temperature $T_{h,i}$, zone air temperature $T_{z,i}$, and lumped wall temperature $T_{w,i}$. The heating system is modeled as an ideal mixed volume, with heat exchange between thermal capacities represented by constant coefficients k. This thermal model for the zones is deliberately kept relatively simple (linear heat transfer, no radiation or infiltration, a single solid capacity). The chosen model is sufficient to create dynamic thermal loads that interact with the shared hydraulic system. A more detailed thermal model per zone would increase the complexity of individual agent NLPs but might obscure the specific challenges arising from the inter-agent coupling, which is central to this study. The control input for each zone i is $u_{p,i}$, representing a normalized mass flow setting for its heating system, ranging from 0 to 1. The thermal zone model incorporates a soft constraint on air temperature $T_{z,i}$ with variable bounds: a lower bound of 17 °C (night setback) and an upper bound of 21 °C (comfort temperature during occupancy). This is implemented using a slack variable $s_{z,i}$ with weight w_i in the agent's objective function, as detailed in Table 2.

For ANN-based modeling, Table 2 specifies the features used to approximate the differential equations. The modeling is autoregressive, i.e. the output variable is always included in the inputs of the ANN. ANN-based approximations are applied only to differential equations, leaving stationary energy balances (e.g., in producers) unchanged from ODE-based modeling. Note that with ANN-based modeling, we skip modeling the heating system, instead considering the upper storage temperature $T_{s,h}$ and mass flow m_i directly in the zone temperature $T_{z,i}$. Consequently, the ANN model for the storage lower layer temperature includes the zone temperature $T_{z,i}$ as a feature and not the heating temperature $T_{h,i}$.

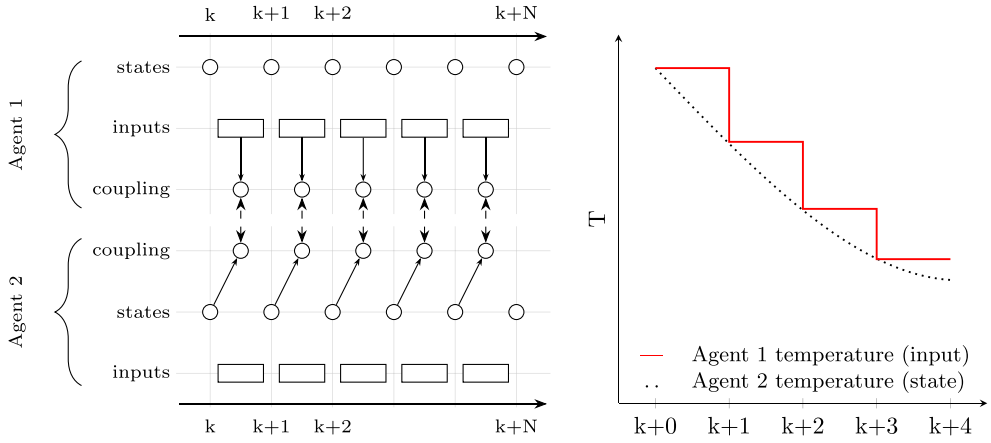
3.3. Challenges in Distributed Integration for NDMPC

Consider the coupling variables defined in Fig. 1 in conjunction with the equations in Table 2. Variables such as the upper layer storage temperature $T_{s,h}$, which effectively serves as the supply temperature for thermal zones, appear in both the storage and consumer subsystems. When solving the optimal control problem for the entire system using a direct transcription method like multiple shooting, an integration scheme is employed to solve the system of differential equations. Depending on the system's stiffness, either a Runge–Kutta method or more sophisticated iterative integration schemes, such as those provided by CVODES in the Sundials Suite [38], can be utilized. These integration schemes consider the simultaneous evolution of states throughout the integration interval, yielding precise solutions to the differential equations.

However, when employing a distributed solution algorithm (see Section 4), typically only discrete values of the coupling variables are exchanged between agents. Consequently, a state like the supply temperature, governed by Eq. (2), is only available at discrete intervals to the thermal zone agents. This effectively reduces the integration order for this variable to one (i.e., an explicit Euler step) in the thermal zone agents. Fig. 2 illustrates this. There are $N+1$ states defined on an equally spaced grid, with N inputs (i.e., controls, disturbances, etc.) defined as constant between two grid points. Consider that variables which serve a role in the local subproblem of an agent (i.e., as state, controlled or non-controlled input) can also be coupling variables. The example in Fig. 2(a) demonstrates the case where the coupling variable serves as an input in Agent 1 (e.g., supply temperature in zone agent), while it is a state in Agent 2 (e.g., upper layer storage temperature in storage agent). To define a consensus constraint suitable for optimization with a distributed optimization algorithm, the dimensions of the coupling variables must match exactly. In this study, we use the input grid (length N) as a coupling grid, although this introduces notable errors, as demonstrated in Fig. 2(b). More sophisticated methods for computing coupling variable values could be considered, such as approximating polynomials and communicating their coefficients

Table 2
Subsystem models, constraints, and objectives for ODE-based and ANN-based MPC.

	Component	ODE-based	ANN-based
Boiler	Model	$T_b = T_{s,c} + u_b \frac{\dot{Q}_{n,b}}{\dot{m}_b c_p}$	Identical to ODE-based
	Constraints	Box constraints on decision vector	
	Objective	$J_b = r_{gas} u_b \frac{\dot{Q}_{n,b}}{\eta_b}$	
CHP	Model	$[\dot{Q}_c, P_{el,c}, \dot{Q}_{gas,c}] = \text{Interp}c(u_c)$ $T_c = T_{s,c} + \frac{\dot{Q}_c}{\dot{m}_c c_p}$	Identical to ODE-based
	Constraints	Box constraints on decision vector	
	Objective	$J_c = r_{gas}\dot{Q}_{gas,c}(u_c) - r_{el} P_{el,c}(u_c)$	
Storage	Model	$\frac{dT_{s,h}}{dt} = f(\dot{m}_i, T_b, T_c, T_{s,h}, T_{s,c}, T_{env})$ $\frac{dT_{s,c}}{dt} = f(\dot{m}_i, T_{h,i}, T_{s,h}, T_{s,c}, T_{env})$ $\forall i \in$ Rooms, see Equations Eqs. (2) to (4)	$T_{s,h}^{k+1} = \mathcal{A}(T_{s,c}^k, T_b^k, T_c^k, \dot{m}_i^k, T_{z,i})$ $T_{s,c}^{k+1} = \mathcal{A}(T_{s,h}^k, T_b^k, T_c^k, \dot{m}_i^k)$
	Constraints	Box constraints on decision vector	Identical to ODE-based
	Objective	–	
Consumer	Model	$\dot{Q}_{h\to z,i} = k_{h,i}(T_{h,i} - T_{z,i})$ $\frac{dT_{h,i}}{dt} = \frac{\dot{m}_{h,i} c_p (T_{s,h} - T_{h,i}) - \dot{Q}_{h\to z,i}}{C_i}$ $\frac{dT_{z,i}}{dt} = \frac{\dot{Q}_{h\to z,i} - k_{z,i}(T_{z,i} - T_{w,i})}{C_{z,i}}$ $\frac{dT_{w,i}}{dt} = \frac{k_{z,i}(T_{z,i} - T_{w,i}) - k_{w,i}(T_{w,i} - T_{amb})}{C_{w,i}}$	$T_{z,i}^{k+1} = \mathcal{A}(T_{w,i}^k, T_{s,h}^k, \dot{m}_i^k)$ $T_{w,i}^{k+1} = \mathcal{A}(T_{z,i}^k, T_{amb}^k)$
	Constraints	$T_{z,\min} \le T_{z,i} + s_{z,i} \le T_{z,\max}$ Box constraints on decision vector	Identical to ODE-based
	Objective	$J_{z,i} = w_i s_{z,i}^2$	



(a) Discretization of variable types (b) Example with temperature

Fig. 2. Mapping of consensus constraints between agents when using direct multiple shooting.

(see [26]), or using a collocation discretization. To avoid large coupling variables, we resort to the unmodified multiple shooting discretization.

Note *We tried communicating the arithmetic mean of the states at $k+1$ and $k+2$ for the interval instead of the flat value at $k+1$, yet noted no significant improvements.*

Concerns about partially reduced integration orders are less relevant for ANN-based modeling. As the ANNs directly map input to a change in output temperature over a specific time interval, they inherently are discrete and do not require an integrator. Since these models are trained using data at the start of the interval, they naturally function with discrete inputs. Of course, without using features from the full system, there is still some information loss, and similar errors can potentially occurs. Still, the ANNs have the potential to mitigate some of these errors by learning some of the associated system behavior, given a suitable feature selection. For our example system, we compared the accuracy of ANN-based modeling with ODE-based distributed integration (see Fig. 4).

3.4. Distributed optimal control problem

The distributed OCP is defined in a general form:

$$\min_{x,u,s,z,c} \quad \sum_{i=1}^{R}\sum_{k=1}^{N} j_i(x_i^k, u_i^k, s_i^k, z_i^k, c_i^k) \tag{5a}$$

$$\text{s.t.} \quad x_{i,0} = x(t=0), \tag{5b}$$

$$x_i^{k+1} = f_i(x_i^k, u_i^k, c_i^k) \qquad \forall i \in \{1,...,R\}, \tag{5c}$$

$$z_i^k = g_i(x_i^k, u_i^k, c_i^k) \qquad \forall i \in \{1,...,R\}, \tag{5d}$$

$$0 \ge h_i(x_i^k, u_i^k, c_i^k) \qquad \forall i \in \{1,...,R\}, \tag{5e}$$

$$\underline{x}_i \le x_i^k \le \bar{x}_i \qquad \forall i \in \{1,...,R\}, \tag{5f}$$

$$\underline{u}_i \le u_i^k \le \bar{u}_i \qquad \forall i \in \{1,...,R\} \tag{5g}$$

The formulation of the DMPC comprises the composition of R agents, each optimizing their objectives j_i over the horizon N, while adhering

Table 3
Overview of the states x, control variables u, algebraic states z, coupling variables c from other systems, slack variables s, for the MPC problem of each subsystem.

	Consumer	Boiler	CHP	Storage
x_i	$T_{z,i}, T_{w,i}, T_{h,i}$ [a]	T_b	T_c	$T_{s,h}, T_{s,c}$,
u_i	$u_{p,i}$	u_b	u_c	–
z_i	$\dot{m}_{h,i}$	$\dot{m}_b$	$\dot{m}_c$	–
c_i	$T_{s,h}$	$T_{s,c}$	$T_{s,c}$	$T_b, T_c, \dot{m}_{h,i}, T_{h,i}$ [a], $T_{z,i}$ [b]
s_i	$s_{T_{z,i}}$	–	–	–

[a] ODE only.
[b] ANN only.

to dynamics defined by f_i and path constraints defined by g_i and h_i (see Table 2). It is important to note that the dynamics function f in this context refers to a discrete mapping $x^k \to x^{k+1}$. This mapping is realized either by integration of the ODEs over the interval or by a forward pass of the ANN, depending on the chosen modeling approach. For the specific system in this study (see Fig. 1 and Table 2), the overall objective function to be minimized, as stated in Eq. (5a), is the sum over the prediction horizon N of the per-step costs j_i for each agent i. The individual agent objectives j_i at each time step k are:

- Boiler agent (j_b): $r_{gas} u_b^k \frac{\dot{Q}_{n,b}}{\eta_b}$
- CHP agent (j_c): $r_{gas}\dot{Q}_{gas,c}(u_c^k) - r_{el} P_{el,c}(u_c^k)$
- Consumer agent $i \in \{1, \dots, N_z\}$ ($j_{z,i}$): $w_i (s_{z,i}^k)^2$
- Storage agent: Has no explicit objective term ($j_s = 0$).

Thus, the total objective summed over all agents for a single time step k is

$$J_{total}^k = \left(r_{gas} u_b^k \frac{\dot{Q}_{n,b}}{\eta_b} \right) + \left(r_{gas}\dot{Q}_{gas,c}(u_c^k) - r_{el} P_{el,c}(u_c^k) \right) + \sum_{i=1}^{N_z} (w_i (s_{z,i}^k)^2) \quad (6)$$

The full OCP objective is then the sum over all N agents $\sum_{k=1}^{N} J_{total}^k$.

Table 3 summarizes the variables of the optimization problems for each agent type in our investigated system. Due to the modeling differences, the variables and couplings are slightly different for the ODE and ANN variants.

4. Algorithms for distributed optimization

To solve the OCP defined in Eq. (5) we consider distributed optimization algorithms. The OCP can be reformulated in a concise way:

$$\min_{x_1,\dots,x_R} \quad \sum_{i \in \mathcal{R}} \tilde{f}_i(x_i) \quad (7a)$$

$$\text{subject to} \quad \sum_{i \in \mathcal{R}} A_i x_i = b, \quad (7b)$$

where $\tilde{f}_i(x_i) = f_i(x_i) + \iota_{\mathcal{X}_i}(x_i)$, with $\iota_{\mathcal{X}_i}(x_i)$ being the indicator function for the local constraint set, defined as:

$$\iota_{\mathcal{X}_i}(x_i) = \begin{cases} 0, & \text{if } x_i \in \mathcal{X}_i, \\ \infty, & \text{otherwise.} \end{cases} \quad (8)$$

$$\left\{ \mathcal{X}_i = x_i \in \mathbb{R}^{n_i} \mid g_i(x_i) = 0, h_i(x_i) \le 0 \right\} \quad (9)$$

Here, $x_i \in \mathbb{R}^{n_i}$ is the local decision vector of subsystem i, $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$ is the local objective function, $A_i \in \mathbb{R}^{m \times n_i}$ is the local coefficient matrix, $b \in \mathbb{R}^m$ is the global resource vector, and g_i and h_i represent the local equality and inequality constraints, respectively. The set $\mathcal{R} = \{1, \dots, R\}$ refers to the agents. A common special case that is included in this problem formulation is the consensus problem, where coupling variables (components of x_i) need to attain an agreed value among a subset of agents $\mathcal{R}_j \subseteq \mathcal{R}$. This is typically formulated such that for each coupling variable, A_i selects the corresponding local variable in agent $i \in \mathcal{R}_j$ (e.g., $A_{i,j\ell} = 1$ for the coupling variable in question, and 0 otherwise for one agent, and $A_{k,j\ell} = -1$ for another agent $k \in \mathcal{R}_j, k \neq i$), and the corresponding entry in b is zero ($b_j = 0$). This ensures $x_{i,\ell} - x_{k,\ell} = 0$. For systems with more than two agents sharing a variable, consensus can be enforced through bilateral constraints (e.g., $x_{i,\ell} = x_{k,\ell}$ and $x_{i,\ell} = x_{p,\ell}$), designating one agent as a "parent" that holds the reference value for that variable, forming bilateral consensus constraints with all other "child" agents sharing this variable. To solve problems defined by Eq. (7), conventional optimization algorithms can be used to solve the full problem at once, or distributed optimization algorithms, that solve the subsystems i in parallel, can be employed. In the following, we briefly cover two distributed optimization algorithms, ADMM and ALADIN.

4.1. Alternating direction method of multipliers (ADMM)

ADMM is a distributed optimization algorithm that solves problem (7) by introducing a central coordinator and local subproblems. The algorithm iteratively updates the local decision vectors, the global resource allocation, and the dual variables. In this work, we use standard ADMM (see [15]) and bilateral coupling to implement consensus constraints. While it is possible to implement a fully decentralized version for consensus ADMM, our implementation includes a central coordinator to keep track of residuals and computation timers. The steps of ADMM are outlined in Algorithm 1.

Algorithm 1 Parallel ADMM for problem (7)

1: Initialize $z^0, \lambda^0, \rho > 0, k_{\max}, r_{\text{tol}}, k = 0$
2: **while** $k < k_{\max}$ **and** $r^k > r_{\text{tol}}$ **do**
3: $\quad x_i^{k+1} \leftarrow \text{argmin}_{x_i} \tilde{f}_i(x_i) + \lambda^{k\top} A_i x_i + \frac{\rho}{2} ||A_i x_i - z^k||^2, \quad \forall i \in \mathcal{R}$ (parallel)
4: $\quad z^{k+1} \leftarrow \text{argmin}_z \sum_{i=1}^{R} A_i x_i - z - b$ (centralized)
5: $\quad \lambda^{k+1} \leftarrow \lambda^k + \rho \sum_{i=0}^{R} A_i x_i - z^{k+1}$ (centralized)
6: $\quad r^k = ||A_i x_i^{k+1} - z^{k+1}||$
7: $\quad k \leftarrow k + 1$
8: **end while**

In Algorithm 1, $z^k \in \mathbb{R}^m$ is the global resource allocation at iteration k, $\lambda^k \in \mathbb{R}^m$ is the dual variable associated with the consensus constraint (7b), and ρ is the penalty parameter for the augmented Lagrangian term. r^k denotes the primal residual, r_{tol} and $k_{\max}$ denote the tolerance on the primal residual and the maximum number of iterations respectively. We do not use any heuristics for dynamic scaling or updates of the penalty parameter ρ, as results during initial testing were not promising.

4.2. Augmented Lagrangian Alternating Direction Inexact Newton (ALADIN)

ALADIN is a distributed optimization algorithm that incorporates second-order information and can find local solutions of non-convex problems [20,39]. Inspired by Sequential Quadratic Programming methods, it introduces a quadratic approximation of the objective function and a linearization of the constraints in the central step. The steps of ALADIN are summarized in Algorithm 2.

Table 4 shows additional symbols that are used in Algorithm 2.

4.3. Practical adjustments and parameters

Several adjustments and heuristics are implemented to enhance ALADIN's convergence and closed-loop performance:

- **Hessian regularization:** Hessian regularization is applied to the hessian approximation B_i to ensure positive definiteness, flipping the sign of negative eigenvalues and putting zero eigenvalues on a small parameter δ_{reg} (see [39]).

Algorithm 2 Basic ALADIN for problem (7)

1: Initialize $z_i^0 \in \mathbb{R}^{n_i}, \lambda^0 \in \mathbb{R}^m, \Sigma^0 > 0, \rho^0 > 0, \mu^0 > 0, \rho_{\max}, \mu_{\max}$
2: **while** $k < k_{\max}$ **and** $r^k > r_{\text{tol}}$ **do**
3: Solve local objectives (parallel)

$$x_i^{k+1} \leftarrow \underset{x_i}{\operatorname{argmin}}\ \tilde{f}_i(x_i) + \lambda^{k\top} A_i x_i + \frac{\rho^k}{2}||x_i - z_i^k||^2,\ \forall i \in \mathcal{R}$$

4: Determine active set and effective equality constraints $\tilde{g}_i(x_i^{k+1})$ (parallel):

$$\tilde{g}_i(x_i^{k+1}) = \begin{bmatrix} g_i(x_i^{k+1}) \\ h_i^a(x_i^{k+1}) \end{bmatrix}$$

where $h_i^a(x_i^{k+1}) = \{h_{i,j}(x_i^{k+1}) : h_{i,j}(x_i^{k+1}) \geq -\epsilon_{act}\}$

5: Compute Sensitivities (parallel):

$\nabla f_i(x_i^{k+1})$ (Gradient)
$B_i^k \approx \nabla^2_{x_i x_i}\left(f_i(x_i^{k+1}) + \gamma_i^\top g_i(x_i^{k+1}) + \mu_i^\top h_i(x_i^{k+1})\right)$ (Hessian approx.)
$\nabla \tilde{g}_i(x_i^{k+1})$ (Constraint Jacobian)

6: Solve the coordination QP (central):

$$\begin{aligned} \Delta x^k = \underset{\Delta x, s}{\operatorname{argmin}}\quad & \sum_{i\in\mathcal{R}} \frac{1}{2}\Delta x_i^\top B_i^k \Delta x_i + \nabla f_i(x_i^{k+1})^\top \Delta x_i \\ & + \lambda^{k\top} s + \frac{\mu^k}{2}||s||^2 \\ \text{s.t.}\quad & \sum_{i\in\mathcal{R}} A_i(x_i^{k+1} + \Delta x_i) = b + s \quad | \lambda_{QP} \\ & \nabla \tilde{g}_i(x_i^{k+1})\Delta x_i = 0, \quad \forall i \in \mathcal{R} \end{aligned}$$

7: Update local targets z_i, global multiplier λ and determine residual r

$$z_i^{k+1} = x_i^{k+1} + \alpha^k \Delta x_i^k$$
$$\lambda^{k+1} = \lambda^k + \alpha^k(\lambda_{QP} - \lambda^k)$$
$$r^k = ||A_i x_i^{k+1} - b||$$

8: **end while**

Table 4
Symbols used in Algorithm 2.

Symbol	Description
γ_i, μ_i	Dual variables associated with local equality and inequality constraints
$\tilde{g}_i(x_i)$	Local equality constraints and active inequalities
$\Delta x^k \in \mathbb{R}^n$	Primal update direction at iteration k
$s \in \mathbb{R}^m$	Slack variable on coupling constraints
$\alpha^k \in (0, 1]$	Step size for updating the primal variables at iteration k
$\mu^k > 0$	Penalty parameter for the slack variable at iteration k
$\rho_{\max}, \mu_{\max}$	Maximum values for ρ and μ
λ_{QP}	Lagrange multipliers from the coordination QP

- **Parameter updates:** ρ is updated according to $\rho^{k+1} = \rho^k \cdot \rho_{update}$.
- **Variable scaling:** We scale coupling variables using factors T_{scale} and $\dot{m}_{scale}$ to improve problem conditioning.
- **Step size control:** We limit the step size α based on the infinity norm of the QP step $\|\Delta x\|_\infty$ and an absolute maximum Δx_{max}, preventing excessively large steps.
- **Best solution recovery:** To deal with instability between iterations, we track and return the best solution based on coupling constraint violation.

The Hessian regularization and ρ-updates are included with ALADIN-α [39], while scaling implementation, step size control and solution recovery were adjusted based on own experience.

5. Reference controllers

To evaluate the performance of the DMPC strategies, we compare them against a conventional baseline controller, and a centralized benchmark MPC, serving as the upper bound. The reference controllers are explained in the following sections.

Rule-based control strategy

The rule-based control strategy serves as a baseline for comparison with the MPC approaches. This control scheme combines PID controllers for local zone temperature regulation with supervisory logic for producer coordination. Individual PID controllers regulate the mass flow through each zone's heating system ($u_{p,i}$) to track time-varying temperature setpoints. The control action is bounded between 0 and 1, representing the normalized pump speed.

During occupied hours (8:00–18:00), the setpoint is 21 °C, while during unoccupied hours, it reduces to 17 °C for energy savings. To ensure comfort at the start of occupancy, the controller initiates a pre-heating period four hours before the scheduled occupancy time, raising the setpoint to 21 °C in advance.

A hysteresis-based strategy manages the storage tank's hot layer temperature to follow a setpoint of 55 °C. The boiler activates when the storage temperature drops below 52 °C (3 K below the 55 °C setpoint) and deactivates when it rises above 55.5 °C (0.5 K above the setpoint). This asymmetric hysteresis band prevents excessive cycling while ensuring adequate heat supply.

Producer prioritization follows a simple hierarchy: the CHP unit operates as the base-load supplier, while the boiler provides peak-load coverage. When the boiler is active due to low storage temperatures, the CHP is commanded to full power (100%) to maximize its contribution and compensate gas consumption through electricity generation. The boiler is then modulated with a PID controller to track the storage temperature setpoint. When the boiler is inactive, the CHP modulates its output through a separate PID controller.

The PID parameters were manually tuned with simulations to provide stable operation. The main design choices for this control are the pre-heating time and the storage temperature supply temperature. We chose the pre-heating time high enough to avoid most comfort violations, and the storage supply temperature based on original considerations when designing the example system. Coincidentally, the MPC simulations finish at a similar storage temperature, minimizing the error when comparing the controls (due to heat that was generated, but is still available in the storage to a different extend).

We would like to note that this is a very strong conventional baseline control, tuned through simulations, similar to how the MPC parameters were also extensively tuned (see Section 6.3). In a real setting, this conventional control would likely perform worse, as pre-heating times cannot be known in advance and PID parameters are not optimal.

Centralized MPC benchmark

The centralized MPC serves as an upper performance benchmark. It utilizes the exact ODE-based process model (Table 2) for all agents. The OCP is transcribed using direct multiple shooting with a CVODES integrator, ensuring full integration order for all states, including those involved in inter-agent coupling, by solving the entire system model simultaneously. The resulting NLP is solved using IPOPT. This setup represents the best achievable performance for an MPC under ideal modeling and centralized information with the given prediction horizon. The current horizon of 6 h (Table 5) does not allow the MPC to foresee the start of the next day's occupancy period when making decisions towards the end of the current day. While a longer horizon could potentially improve benchmark performance further, it was kept consistent with the DMPC setups, where longer horizons would significantly increase computational demands and exacerbate challenges related to ANN model accuracy over extended predictions.

6. Results

In this section, we present numerical results for the control of the example building energy system described in Section 2.3. We evaluate various DMPC configurations, featuring both ODE-based and ANN-based models, and compare the distributed optimization algorithms ADMM and ALADIN. The distributed solvers (ADMM and ALADIN) utilize IPOPT to solve local subproblems. All simulations are implemented in MATLAB using a modified version of the ALADIN-α Toolbox [39]. Parameter tuning for the distributed algorithms is orchestrated using Weights & Biases [40], running with matlabengine from Python. Simulations are conducted on a machine with an Intel(R) Xeon(R) Silver 4216 processor. ANN models are trained with Keras [41] in Python. The subsequent subsections detail the closed-loop control performance, the accuracy of distributed integration methods, the parameter tuning process, and a closed-loop scalability analysis of the system.

6.1. Closed-loop control

To validate the efficacy of the proposed DMPC schemes, we conducted a series of closed-loop simulations on the example system comprising two thermal zones (five agents in total, as per Fig. 1). The performance of these schemes is compared against the conventional rule-based controller and the centralized MPC benchmark. For the DMPC methods, we present results from ALADIN and ADMM using ANN-based process models, specifically showcasing the best-performing configurations identified during the parameter tuning process detailed in Section 6.3.3. Table 5 outlines the key simulation parameters used for these comparisons. Controller performance is quantified by the total energy cost in Euro and the RMSCV in Kelvin. The RMSCV is calculated similarly to the Root Mean Square Error but only considers deviations below the lower comfort bound; if the temperature is above the comfort bound, the comfort violation is zero.

Fig. 3 illustrates a 48-h closed-loop simulation, comparing the conventional controller, the centralized benchmark MPC, and the best-tuned ANN-based ALADIN and ADMM controllers. The centralized benchmark achieves the best performance with an RMSCV of 0.01 K and operation costs of 421 €. The conventional controller performs well with negligible comfort violations (RMSCV: 0.05 K) and operation costs of 440 €. The ALADIN-based DMPC achieves comparable performance (RMSCV: 0.08 K, costs: 431 €), with operating costs slightly higher than the benchmark, and slightly lower than the conventional control. The ADMM-based approach shows the highest costs (486 €) with an RMSCV of 0.07 K. The ALADIN-based DMPC exhibits more pronounced oscillations in control inputs, particularly the valve openings. This behavior can be attributed to potential inaccuracies in the ANN models or suboptimal convergence characteristics of the distributed algorithm, which in turn cause the controller to make strong corrective actions. In a practical setting, such oscillations can be mitigated by adding penalties for rapid control changes, further tuning the cost function, or improving model accuracy. Nevertheless, the key takeaway is that ALADIN generally identifies the optimal operational strategy but may lack precision in execution due to these factors. In contrast, the ADMM-based DMPC performs noticeably worse, particularly by failing to effectively utilize the night setback strategy, leading to higher operational costs. It is important to note that this comparison is quite favorable to the conventional controller. Its strong performance is largely due to careful tuning in simulation. In a scenario with more complex dynamics, such as highly nonlinear producer efficiencies (e.g., heat pumps benefitting from precise supply temperature control) or variable energy tariffs, the advantages of MPC-based approaches would likely be more pronounced.

Table 5
Simulation parameters.

Parameter	Value
Simulation time	48 h
Max ALADIN/ADMM iterations	50
ALADIN/ADMM tolerance r_{tol}	1e−3
MPC step size	30 min
Prediction horizon	12 steps (6 h)

Table 6
Parameter tuning overview.

Symbol	Algorithm	Range	Description
ρ_0	Both	[1e−4, 1e6]	Initial penalty parameter
T_{scale}	Both	[0.1, 1e3]	Temperature scaling factor
$\dot{m}_{scale}$	Both	[0.1, 1e2]	Mass flow scaling factor
ρ_{update}	ALADIN	[1, 2]	Penalty parameter update factor
ρ_{max}	ALADIN	[1e5, 1e9]	Maximum penalty parameter
μ_0	ALADIN	[1e−2, 1e6]	QP penalty parameter
α	ALADIN	[0.4, 1.0]	Absolute step size limitation
regParam	ALADIN	[5e−5, 3e−3]	Hessian regularization parameter
actMargin	ALADIN	[−1e−3, −5e−6]	Margin for active set detection
max_iter	ADMM	[15, 400]	IPOPT max iterations
tol	ADMM	[1e−4, 1]	IPOPT acceptable tolerance[a]
compl_tol	ADMM	[1e−5, 1]	IPOPT complementarity tolerance[a]
constr_tol	ADMM	[1e−5, 1]	IPOPT constraint violation tolerance[a]

[a] Referring to acceptable tolerance of IPOPT.

6.2. Distributed Integration Accuracy of ANN s and ODEs

In Section 3.3, we outlined how integration accuracy can be compromised in a distributed setting. Here, we compare the error propagation of distributed integration (with fixed neighbor states over the interval) and ANN-based discrete modeling against the benchmark of full system integration.

We conducted 1000 simulations with random initial conditions and inputs, comparing the errors at 3-h and 12-h endpoints to the full integration solution. The step size for constant inputs is set at 30 min, matching the MPC step size used in our study. Fig. 4 illustrates the results. The top row of Fig. 4 displays the prediction error in room temperature (T_z), while the bottom row shows the prediction error in heating return temperature (T_h). In all scenarios, distributed modeling leads to an underestimation of final temperatures, with distributed ODEs exhibiting significantly larger errors. It is evident that for the energy system in this study, some form of error mitigation technique is necessary to achieve performance comparable to centralized MPC. While Fig. 4 shows better accuracy for ANNs, the improvements in closed loop control are marginal (see Fig. 8). While this work does not offer a concrete solution to this problem, we want to emphasize its importance, as it is not commonly noted in the literature.

6.3. Parameter tuning

To optimize the performance of the ADMM and ALADIN algorithms, we conducted extensive parameter tuning using Weights & Biases (wandb) sweeps. This approach allowed us to systematically explore the parameter space and identify optimal configurations for both algorithms. Table 6 shows the parameters that are optimized for the algorithms.

6.3.1. Tuning process

Two of the most impactful parameters are T_{scale} and $\dot{m}_{scale}$, which are used to scale the problem. While the ALADIN as proposed in i.e. [17] incorporates a scaling matrix Σ in the local coupling term, we found scaling the variables themselves, affecting the entire OCP, to be more effective. Scaling factors are implemented like this:

$$\tilde{T} = \frac{T}{T_{scale}}, \tag{10}$$

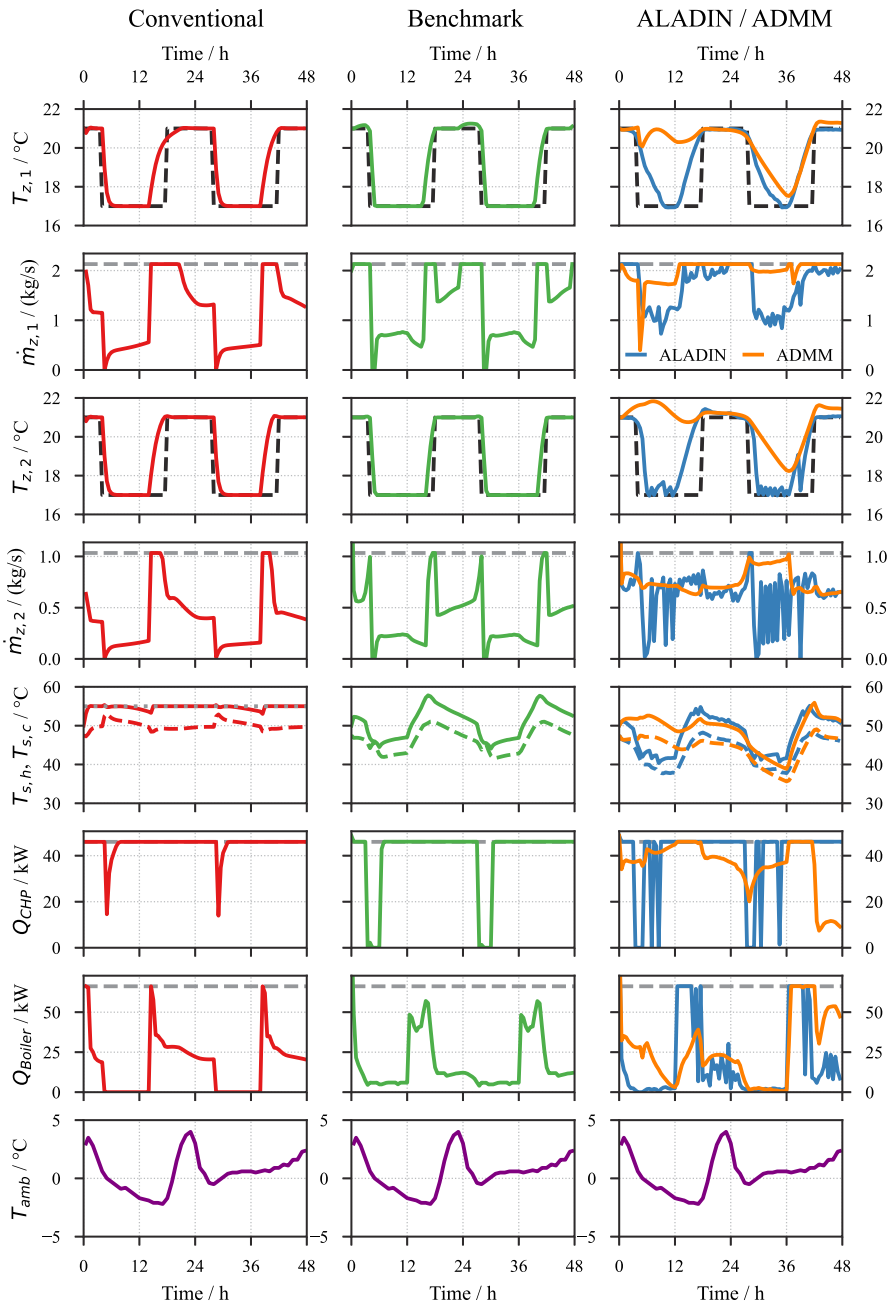


Fig. 3. Comparison of conventional controller, benchmark central MPC and DMPC with ADMM and ALADIN over a 48-h closed-loop simulation period.

where $\tilde{T}$ the optimization variable instead of T. For example, if the original optimal value for a temperature in the OCP was 300 K, and the scaling factor T_{scale} is 100 K, the optimum returned from the solver would be 3. This results in well-scaled variables for the coordination operations in ADMM and ALADIN. We scale all temperatures and all mass flows in the OCP with the same factor respectively.

For ADMM, we tune some solver parameters for the local solution with IPOPT, as preliminary tests showed a high sensitivity of ADMM to the solution tolerance of the local problems (where more precise was not necessarily better). For ALADIN, we chose suitable IPOPT tolerances beforehand, but excluded them from the large-scale tuning, as we found ALADIN to be less sensitive in that regard.

We employed both Bayesian optimization and random search to explore the parameter space defined in Table 6. The optimization was evaluated by a custom cost function J_{tuning}, defined as:

$$J_{\text{tuning}} = J_{\text{econ}} + P_{\text{cons}} \tag{11}$$

where J_{econ} is the economic cost function of the MPC, and P_{cons} is a penalty term for the consensus constraint violations r_p.

$$P_{\text{cons}} = \begin{cases} \|r_p\|^3 & \text{if } \|r_p\| > 10^{-3} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

This formulation ensures that solutions with small consensus violations (less than 10^{-3}) are compared based on the lowest reached cumulative objective, while larger violations incur a cubic penalty, strongly encouraging the optimizer to find solutions that satisfy the consensus constraints. An artificial high value for J_{tuning} is set, when the optimization hits a time limit of 200 s. The search ranges for each parameter were chosen based on preliminary experiments and domain knowledge, as shown in Table 6. Log-uniform distributions were used for parameters spanning several orders of magnitude, while uniform distributions were used for parameters with narrower ranges.

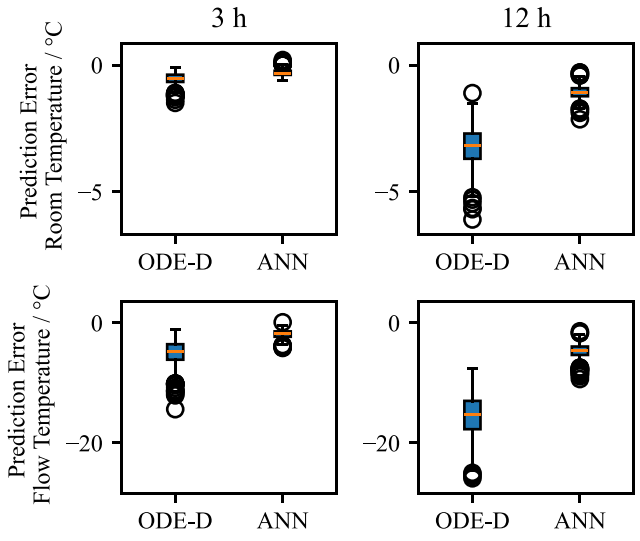


Fig. 4. Box plot for the comparison of integration error propagation for ANNs and ODEs with reduced neighbor integration order over 1000 random sample simulations.

6.3.2. Tuning results and analysis

To visualize the results of our parameter tuning efforts, we have created a cumulative distribution function (CDF) plot of the tuning fitness function J_{tuning} for both ADMM and ALADIN algorithms, comparing random search and Bayesian optimization approaches. This plot is presented in Fig. 5.

The optimal value of the tuning function, which directly corresponds to the DMPC objective function, is approximately 50. Notably, around 40% of ALADIN configurations yield a tuning function value below 60, indicating a high proportion of convergent solutions. In contrast, only about 10% of ADMM configurations converge, despite having fewer parameters to tune. This disparity suggests that ALADIN exhibits greater robustness to parameter variations, with a wider range of parameter sets leading to near-optimal solutions, while ADMM's convergence to the optimum is more sensitive to specific parameter combinations. The figure also reveals a slight advantage in using Bayesian optimization over random search, as evidenced by a marginally higher fraction of convergent solutions among open-loop runs for both algorithms.

Fig. 6 provides deeper insights into the parameter tuning results for ADMM. The parallel coordinates plots visualize the relationship between key parameters (ρ, T_{scale}, and $\dot{m}_{scale}$) and the resulting performance for the best and worst ADMM runs. Examining the best-performing runs in Fig. 6(a) confirms common ADMM intuitions:

- The penalty parameter ρ needs to be sufficiently high to prevent divergence.
- Both $\dot{m}_{scale}$ and T_{scale} should be chosen so that the scaled variables are close to 1.

Configurations that fail to meet these conditions consistently result in poor ADMM performance.

However, Fig. 6(b) reveals that adhering to these general guidelines does not guarantee optimal performance. Poor runs can still occur even with large values of ρ and appropriate scaling factors. This suggests that while these intuitive guidelines are necessary for good performance, they are not sufficient on their own for this challenging, non-convex problem.

For ALADIN, the tuning results are shown in Fig. 7. Compared with ADMM, it is apparent that the best runs (Fig. 7(a)) make up a larger portion of overall runs compared to the poor runs (Fig. 7(b)). Additionally, only the temperature scaling T_{scale} has a clearly visible impact, which can also be seen for the worse runs, where low values for T_{scale} are clearly correlated with high (i.e. poor) values for the fitness function. The mass flow scaling $\dot{m}_{scale}$ seems to matter less, while the penalty parameter ρ is more consistent at lower values. However, if ρ is too small, the algorithm will diverge fully, which is not shown here to improve scaling of the figure. Summarizing, ALADIN demonstrates more consistent good performance when T_{scale} and ρ are chosen within intuitively good ranges compared to ADMM.

While these findings indicate some correlation between intuitive parameter choices and actual performance, sophisticated parameter tuning is still required for either algorithm to achieve optimal performance. In the following section, we examine how a good open-loop performance translates to closed-loop performance.

6.3.3. Transfer to closed-loop performance

To assess whether open-loop tuning translates to effective closed-loop performance, we executed the top 10 configurations from the parameter tuning for each DMPC scheme. Fig. 8 presents the closed-loop results of all examined DMPC schemes for 48-h simulations. The x-axis represents operating costs, while the y-axis shows comfort violation (RMSCV). The benchmark occupies the bottom-left corner, indicating the least comfort violation and lowest operating costs. Judging from the benchmark and detailed plots of other runs, we consider runs with RMSCV below 0.2 K to be good, usually experiencing one smaller deviation in the room temperature profile. Runs with an RMSCV above 0.4 K usually have serious issues, disqualifying them considering the expectations one has on a sophisticated control scheme like MPC. Table 7 summarizes the key performance indicators presented in Fig. 8.

IPOPT results are consistently close to the benchmark, regardless of the modeling approach, though still inferior, highlighting the numerical errors introduced through the reduced integration order of neighboring states, or the ANN models. Note that the IPOPT results are obtained with the variable scaling from the ADMM-tuning, explaining differences in operation cost and execution time. ADMM results exhibit good comfort levels at high and widely varying operating costs. These are often the result of serious overheating, disregarding the temperature profile, and low sensitivity towards choosing the correct producer. ALADIN results with ANNs show acceptable comfort violations and lower operating costs compared to ADMM, though still notable variation exists. Comparing ALADIN results to IPOPT in detailed simulations reveals very similar control profiles. The higher RMSCV is usually attributed to comfort violations around the edges of the occupancy window, i.e. late heating or early stopping. ALADIN results with ODEs are significantly worse, displaying unacceptable control behavior, often running into the iteration limit without finding a convergent solution. This could be attributed to the tuning being optimized for ANNs, with the computational intensity of ODEs making ODE-targeted tuning impractical.

Regarding computation times, shown on the right side of Fig. 8, ANN-based IPOPT is the fastest, requiring under five seconds per MPC step, followed by the benchmark. ANN-based ALADIN, ANN-based ADMM, and ODE-based IPOPT are comparable, taking between 40–80 s. ODE-based ADMM and ALADIN are notably slower, requiring around 500 s per optimization.

While these optimization times may seem large, several factors contribute to this. CVODES' inherent slowness affects all ODE-based approaches. Additionally, ALADIN and ADMM do not scale efficiently for small systems (only 5 agents in this case) and require multiple IPOPT problem solutions. Furthermore, CasADi's C-code generation feature was not utilized in this study, as it cannot be used for code including CVODES.

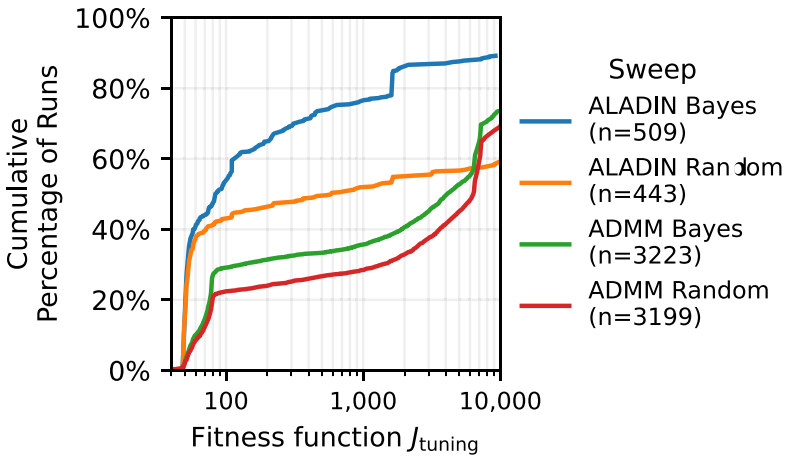
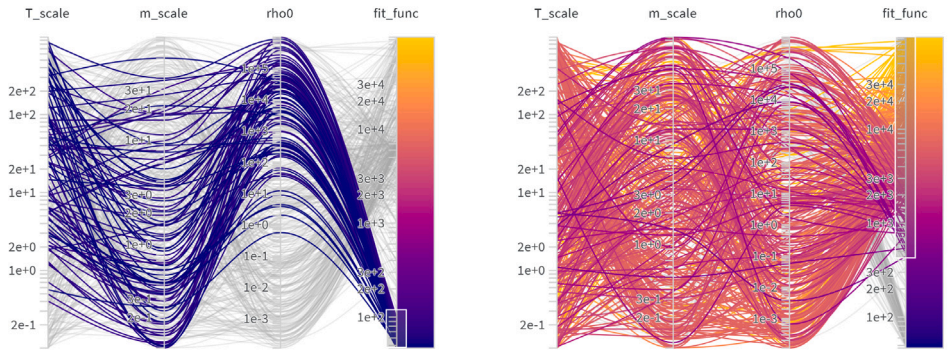
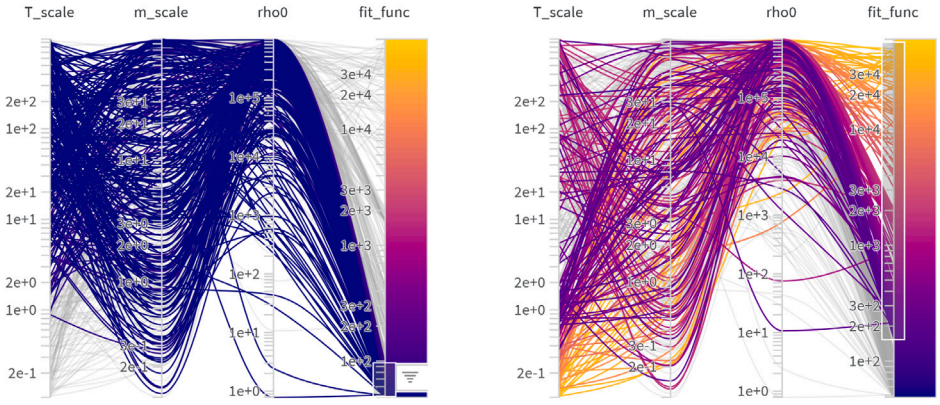


Fig. 5. Cumulative distribution function of the tuning fitness function J_{tuning} for ADMM and ALADIN algorithms, comparing random search and Bayesian optimization approaches.



(a) Highlighting the best runs (b) Highlighting worse runs

Fig. 6. Parallel coordinates plots highlighting the best and worst ADMM runs with respect to their tuning values for ρ, T_{scale}, and $\dot{m}_{scale}$.



(a) Highlighting the best runs (b) Highlighting worse runs

Fig. 7. Parallel coordinates plots highlighting the best and worst ALADIN runs with respect to their tuning values for ρ, T_{scale}, and $\dot{m}_{scale}$.

6.4. Closed-loop scalability study

To evaluate performance with larger systems, we progressively increased the number of consumers. This scaling was performed while keeping storage and producer parameters constant; consumer size was scaled down proportionally to the number of systems. For scenarios with five or more consumers, we generated three randomized room sizes, training a new ANN as a process model for each of these configurations. We tested systems with 2, 5, 10, 20, and 40 consumers, conducting closed-loop simulations over a 24-h period using the best configuration determined from open-loop parameter tuning. Due to the

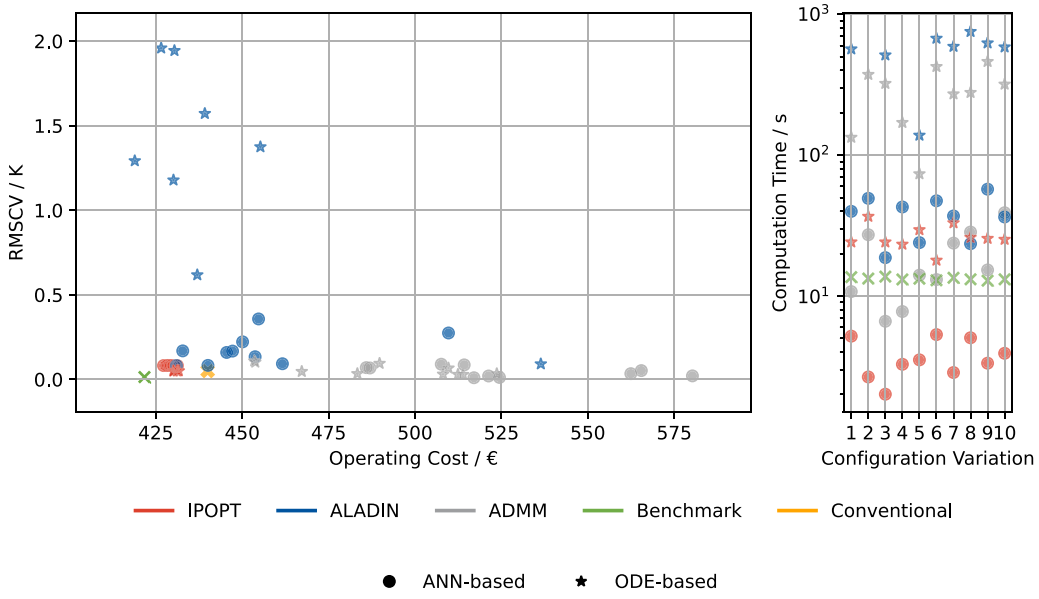


Fig. 8. Comparison of DMPC schemes: Operating costs vs. Comfort violation (RMSCV) and Computation times per MPC step.

Table 7
Summary of controller performance metrics from closed-loop simulations (48-h period, 2 thermal zones). Values for ADMM and ALADIN represent averages over the top 10 tuned configurations.

Solver	Modeling	Operating cost (€)			RMSCV (K)			Time (s)
		Avg.	Min.	Max.	Avg.	Min.	Max.	Avg.
IPOPT	ODE[a]	421.71			0.01			13.25
IPOPT	ODE-D	430.85	430.05	431.67	0.05	0.05	0.05	26.55
IPOPT	ANN	429.51	427.22	431.25	0.08	0.08	0.08	3.71
ADMM	ODE-D	491.54	453.50	523.67	0.06	0.03	0.11	282.84
ADMM	ANN	526.57	485.89	580.27	0.05	0.01	0.09	18.59
ALADIN	ODE-D	446.72	418.91	536.40	1.25	0.09	1.96	556.06
ALADIN	ANN	452.64	430.97	509.69	0.17	0.08	0.36	37.63

[a] This row (IPOPT, ODE) represents the benchmark (see Fig. 8). The key difference in ODE-D are the distributed integration errors introduced, see Section 3.3.

large runtimes of ODE-based approaches, we only performed these simulations with the ANN-based MPC. Fig. 9(a) illustrates the performance of DMPC for increasing numbers of consumers comparing IPOPT with ADMM and ALADIN.

All methods exhibit a significant increase in runtime as the number of consumers grows. Beyond 20 consumers, ADMM and ALADIN outperform IPOPT in terms of computational speed. ADMM reaches its iteration limit after 20 consumers, a behavior not observed with ALADIN in the tested scenarios. Operational costs and comfort violations are elevated and generally more volatile for cases with more than two consumers, suggesting less optimal solutions. This trend is consistent across all algorithms, including the central solution provided by IPOPT. A plausible explanation is the decreased model accuracy of the ANNs, as a new set of ANNs is trained for each case.

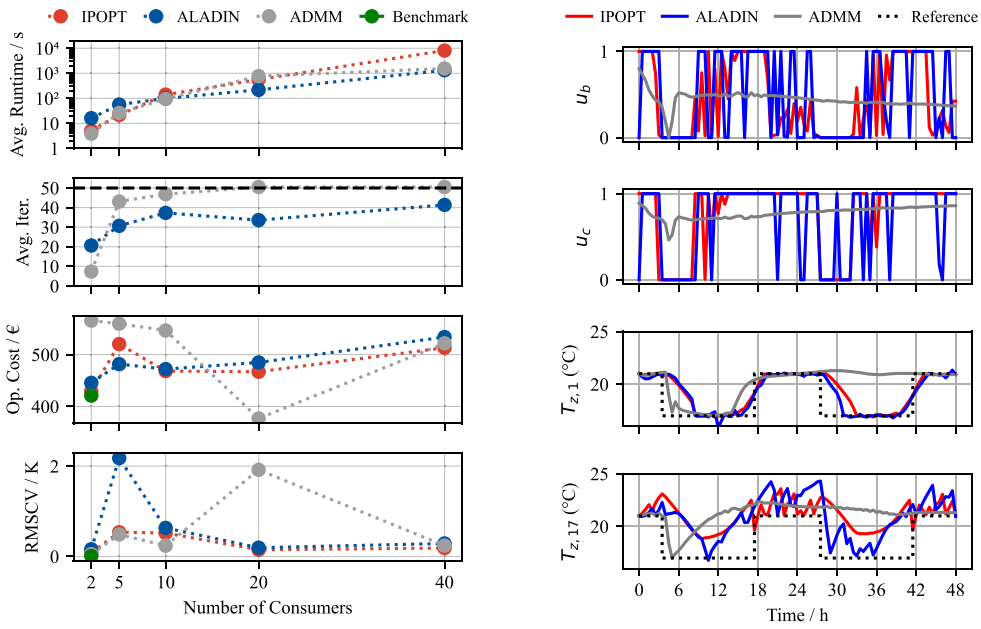
While the KPIs (Key Performance Indicators) in Fig. 9(a) might suggest that all algorithms converge to similar solutions for 40 consumers, Fig. 9(b) reveals substantial differences in operational detail between the algorithms. The MPC using IPOPT and ALADIN adheres to the reference lower bound and deactivates producers at the start of the night. Performance varies across rooms; some consumers (e.g., zone 1) exhibit smooth operation, while others show more erratic behavior. At this stage, it is unclear whether these variations stem from model inaccuracies or insufficient convergence of the algorithms. ADMM, conversely, produces a feasible but clearly suboptimal operation, disregarding night setbacks and continuously operating the boiler. Consequently, while ALADIN still has areas requiring refinement for real-world application, it achieves a solution much closer to the optimum than ADMM.

Although the DMPC is an order of magnitude faster than the central MPC for the largest case (40 consumers), the runtimes remain substantial. We therefore conducted a more in-depth examination of the computational profile. Fig. 10 presents a detailed breakdown of ALADIN's computation times.

ALADIN's computation comprises the solution of local NLPs, calculation of local gradients, Hessian regularization, and the central QP. As steps 1–3 can be executed in parallel for all agents, we only consider the slowest agent for those components. Local NLP solution time is in all cases the dominant component of the computation, as shown in Fig. 10(a). It remains relatively constant between 2 and 5 consumers but increases exponentially thereafter. The central QP solution scales exponentially from the outset, becoming a significant contributor to the overall computation time as the number of consumers increases. Hessian regularization and sensitivity calculations also scale, but have a minor impact on total time.

A closer examination of local NLP solution times in Fig. 10(b) reveals the average NLP solution time for a single iteration by subsystem type (consumer, producer, storage). While NLP solution time is independent of system size for consumers and producers, it increases with the number of consumers for the storage subsystem. This explains the minimal change in runtime between 2 and 5 consumers, as the dominant system shifted from consumer to storage.

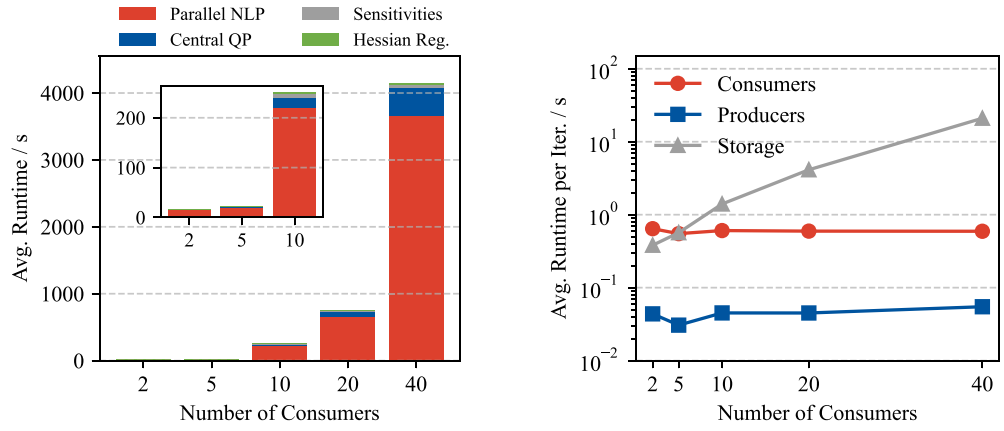
The storage NLP scales in size with the number of consumers due to coupling variables affecting system dynamics for each consumer. Consequently, the advantage of DMPC is diminished, as a single agent within the network still scales with the overall problem size. It is also noteworthy that the central QP becomes a significant portion of



(a) Closed-loop performance metrics for increasing numbers of consumers.

(b) 40 consumer closed-loop simulation results of relative boiler power u_b, CHP power u_c, and exemplary operation for zones 1 and 17.

Fig. 9. Performance analysis for larger systems in closed-loop operation.



(a) Breakdown of the computation time required by steps of ALADIN

(b) Average time for NLP solution of each subsystem type per iteration

Fig. 10. Analysis of the ALADIN computation times for varying consumer numbers.

optimization time for larger systems, indicating a potential bottleneck for ALADIN.

7. Discussion

In the following section, we summarize our findings, disclose limitations of this study and consider future research directions.

7.1. Findings regarding NDMPC in Building Energy Systems

This study advances practical understanding of NDMPC for buildings regarding system topology, optimization algorithms and transcription and modeling.

Performance and scalability. The results demonstrate that NDMPC can achieve performance levels comparable to centralized MPC in building energy management. This finding is particularly significant for large-scale systems, where a parallel implementation shows computational advantages over centralized approaches, disregarding communication overhead. However, it is crucial to consider the system architecture carefully. Star-like graphs with an agent sharing many individual consensus constraints with neighbors should be avoided, as that can lead to exponential growth in computational complexity for that agent, negating the scaling advantages of distributed architectures. This phenomenon was also observed in the study [8], although less severe, as their storage modeling is linear. While in this study, distributed approaches outperformed the central solution computationally, this only occurred for system sizes where computation times and solution quality are insufficient for all approaches, prompting a need for further

improvements before these technologies become viable. We also noticed significant sensitivity towards algorithm parameters and system configuration (as seen in Fig. 9).

Algorithm characteristics and tuning. Our investigation revealed distinct characteristics of different algorithms. ADMM proved effective at finding approximate and feasible solutions, but severely lacked in achieving energy optimality compared to ALADIN or a central solver like IPOPT. ALADIN in general converged to higher accuracies, providing energy-efficiency close to IPOPT in many cases, although its solutions are less stable, resulting in violations of comfort constraints from time to time. Both ADMM and ALADIN may fail to converge without sophisticated parameter tuning and variable scaling, but ALADIN is less sensitive to variations in variable scaling. While open-loop tuning can effectively identify non-convergent configurations, we noted significant and seemingly random variations in performance among convergent setups. This variability highlights the complexity of NDMPC systems and the challenges in predicting and optimizing their performance across different scenarios.

Modeling and transcription strategies. The study highlighted the significant impact of modeling and OCP transcription choices on DMPC performance. Distributed multiple shooting transcriptions that only consider neighboring states as piece-wise constant can introduce significant errors, reducing controller performance. Additionally, using robust integrators like CVODES results in prohibitive computation times for systems that require them. NARX models like ANNs alleviate these issues, showing major improvements in computation time while gracefully handling these integration errors, as they are inherently trained on discrete data. One should keep in mind that in systems where a Euler-step is sufficient for integration accuracy, these issues are less of a concern.

7.2. Limitations

While our study provides insights into the performance of NDMPC for building energy systems, it is important to acknowledge several limitations:

- The computation time analysis for parallel ADMM and ALADIN implementations does not account for communication overhead. In real-world applications, this overhead can be a significant, if not the main, contributor to the overall run time [6].
- Our overall computation times should be considered in the context of the software environment used. The simulations were conducted using MATLAB with CasADi without C–Code generation, which may not achieve the same performance as directly running optimized C-code with tailored low-level algorithms.
- The performance of artificial neural networks (ANNs) introduces notable variability in closed-loop results, even for centralized MPC. We put significant effort into tuning and feature selection for the two-room case to ensure good performance, however we cannot guarantee that this transfers to the cases with more rooms.
- Creating high-quality ANN models for all subsystems in large-scale applications and automatically validating their performance remains a significant challenge. This limitation becomes particularly pronounced as the system size increases.
- While we tested numerous parameterizations and attempted to treat all algorithms fairly, there remain unexplored options. These include various ADMM modifications and a more detailed analysis of solution accuracy and termination criteria across different algorithms. We also ran the ODE-based configurations with parameters from the open-loop tuning with ANN-based MPC, as the high runtimes of the ODE-based MPC make tuning with them impractical.
- The study uses a real ambient temperature profile and representative building/zone dimensions, but does not incorporate other real-world data sources such as occupancy profiles, internal gains schedules from measurements, or validate against a physical testbed. A comprehensive validation with extensive real-world data, including various uncertainties, is considered out of scope for this paper, which focuses on the algorithmic and numeric aspects of modeling NDMPC for strongly coupled systems.

7.3. Future research directions and potential enhancements

The results of this study show promise for NDMPC and highlight further research directions towards practical and reliable implementation.

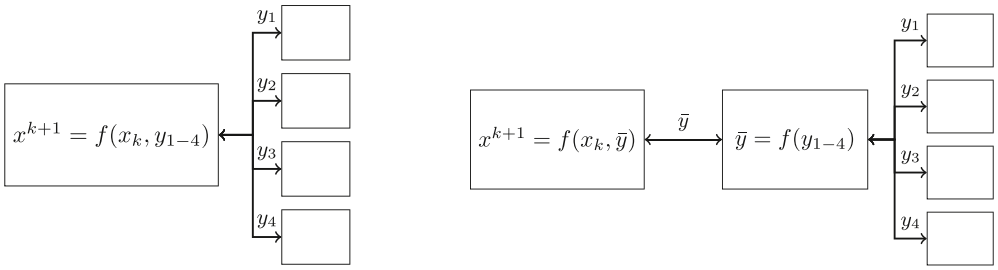
Alternative discrete modeling approaches. While Artificial Neural Networks (ANNs) performed well in our study, their success may be attributed more to their discrete nature than to the specific ANN architecture. Future research could explore other inherently discrete, data-based modeling options that are computationally more efficient and easier to train than ANNs. Hammerstein–Wiener type models, for instance, could offer a promising alternative, potentially combining the benefits of discrete representation with reduced computational complexity and training difficulty.

Comparative analysis of transcription methods. Our current study focused on multiple shooting but did not include a comprehensive comparison with direct collocation methods. In a previous study [35], we explored direct collocation for a similar system, which proved highly effective for centralized MPC, demonstrating a significant improvement in computation time compared to our current centralized benchmark. Nevertheless, the distributed version using direct collocation encountered challenges, requiring an excessive number of ADMM iterations to converge. This difficulty may be attributed to the higher dimensionality of the coupling variables in the collocation approach. It is important to note that the previous study lacked the variable scaling and intensive parameter tuning implemented in our current work, making direct comparisons between the two approaches problematic. These observations underscore the need for further research to fully understand the trade-offs between multiple shooting and direct collocation in NDMPC contexts. Finally, further improvements should be made regarding distributed transcription of OCPs, aiming at reducing numerical errors with regard to neighboring integration order, while keeping coupling variables of low dimension.

Topology optimization for improved scaling. Future work could focus on adjusting the system topology to enhance scalability. One approach is the introduction of aggregator agents at points where numerous coupling variables converge. These aggregator agents, comprising simple equations (for example, computing the total enthalpy flow from a number of mass flows and temperatures), could significantly reduce the number of variables handled by more complex agents, such as storage agents. This strategy has the potential to distribute computational load more evenly and improve overall system performance. Fig. 11 shows how the topology could be changed with an aggregator agent.

8. Conclusion

This study investigates the application of NDMPC to complex BES, focusing on scenarios with multivariate coupling between subsystems. We compared the performance of ADMM and ALADIN optimization algorithms against a centralized IPOPT benchmark, using both ODE-based and ANN-based modeling approaches. Our results demonstrate that NDMPC can achieve performance comparable to centralized MPC, particularly when using ALADIN. We also highlighted the critical role of parameter tuning and variable scaling for both ADMM and ALADIN, observing that ALADIN exhibits greater robustness to parameter variations. Furthermore, we identified a scalability bottleneck in star-like



(a) Original system topology (b) Modified system topology with aggregator

Fig. 11. Comparison of system topologies: (a) shows the original direct connection between the central agent and consumer agent, while (b) introduces an aggregator node that performs light computation to reduce the number of inputs for the central agent.

topologies where a central agent, such as the storage agent in our case study, experiences increasing computational burden with growing system size. The use of ANN-based models offered computational advantages compared to ODE-based approaches. However, further research is needed to address the identified scalability limitations, exploring alternative system topologies and aggregation techniques. Future work should also focus on improving the consistency of the controller, especially with regard to model accuracy and robustness towards parameter variations. It should also include application to a variety of systems, especially ones that bring out the full potential of NMPC, featuring more nonlinear producers such as heat pumps, and varying energy tariffs, further incentivizing utilization of the storage.

CRediT authorship contribution statement

Steffen Eser: Writing – original draft, Visualization, Validation, Methodology, Conceptualization. **Ben Spoek:** Writing – review & editing, Visualization, Software, Methodology, Investigation. **Augustinus Schütz:** Software, Methodology, Investigation. **Phillip Stoffel:** Writing – review & editing, Supervision, Project administration. **Dirk Müller:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used Claude and Gemini in order to improve language and readability. After using this service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We gratefully acknowledge the financial support provided by the BMWK (Federal Ministry for Economic Affairs and Climate Action, Germany), promotional reference 03EN1006A.

Data availability

Data will be made available on request.

References

[1] Killian M, Kozek M. Ten questions concerning model predictive control for energy efficient buildings. Build Environ 2016;105:403–12. http://dx.doi.org/10.1016/j.buildenv.2016.05.034.

[2] Drgoňa J, Arroyo J, Cupeiro Figueroa I, Blum D, Arendt K, Kim D, Ollé EP, Oravec J, Wetter M, Vrabie DL, Helsen L. All you need to know about model predictive control for buildings. Annu Rev Control 2020;50:190–232. http://dx.doi.org/10.1016/j.arcontrol.2020.09.001.

[3] Saloux E, Candanedo JA, Vallianos C, Morovat N, Zhang K. From theory to practice: A critical review of model predictive control field implementations in the built environment. Appl Energy 2025;393:126091. http://dx.doi.org/10.1016/j.apenergy.2025.126091, URL: https://linkinghub.elsevier.com/retrieve/pii/S0306261925008219, Publisher: Elsevier BV.

[4] Müller MA, Allgöwer F. Economic and distributed model predictive control: Recent developments in optimization-based control. SICE J Control Meas Syst Integr 2017;10(2):39–52. http://dx.doi.org/10.9746/jcmsi.10.39.

[5] Maestre JM, Negenborn RR, editors. Distributed model predictive control made easy, In: Intelligent systems, control and automation: Science and engineering, 1st ed. 2014 ed.. Springer Netherlands, Dordrecht: Imprint: Springer; 2014, http://dx.doi.org/10.1007/978-94-007-7006-5, number 69.

[6] Eser S, Storek T, Wüllhorst F, Dähling S, Gall J, Stoffel P, Müller D. A modular Python framework for rapid development of advanced control algorithms for energy systems. Appl Energy 2025;385:125496. http://dx.doi.org/10.1016/j.apenergy.2025.125496.

[7] Shi Y, Tuan HD, Savkin AV, Lin C-T, Zhu JG, Poor HV. Distributed model predictive control for joint coordination of demand response and optimal power flow with renewables in smart grid. Appl Energy 2021;290:116701. http://dx.doi.org/10.1016/j.apenergy.2021.116701.

[8] Lefebure N, Khosravi M, Hudoba de Badyn M, Bünning F, Lygeros J, Jones C, Smith RS. Distributed model predictive control of buildings and energy hubs. Energy Build 2022;259:111806. http://dx.doi.org/10.1016/j.enbuild.2021.111806.

[9] Lin F, Adetola V. Flexibility characterization of multi-zone buildings via distributed optimization. In: 2018 annual American control conference. ACC, Milwaukee, WI: IEEE; 2018, p. 5412–7. http://dx.doi.org/10.23919/ACC.2018.8431400.

[10] Yang Y, Hu G, Spanos CJ. HVAC energy cost optimization for a multizone building via a decentralized approach. IEEE Trans Autom Sci Eng 2020;17(4):1950–60. http://dx.doi.org/10.1109/TASE.2020.2983486.

[11] Rawlings JB, Angeli D, Bates CN. Fundamentals of economic model predictive control. In: 2012 IEEE 51st IEEE conference on decision and control. CDC, IEEE; 2012, p. 3851–61. http://dx.doi.org/10.1109/CDC.2012.6425822.

[12] Christofides PD, Scattolini R, Muñoz de la Peña D, Liu J. Distributed model predictive control: A tutorial review and future research directions. Comput Chem Eng 2013;51:21–41. http://dx.doi.org/10.1016/j.compchemeng.2012.05.011.

[13] Gabay D, Mercier B. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. Comput Math Appl 1976;2(1):17–40. http://dx.doi.org/10.1016/0898-1221(76)90003-1.

[14] Glowinski R. On alternating direction methods of multipliers: A historical perspective. In: Fitzgibbon W, Kuznetsov YA, Neittaanmäki P, Pironneau O, editors. Modeling, simulation and optimization for science and technology. vol. 34, Dordrecht: Springer Netherlands; 2014, p. 59–82. http://dx.doi.org/10.1007/978-94-017-9054-3_4.

[15] Boyd S, Parikh N, Chu E, Peleato B, Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends® Mach Learn 2011;3(1):1–122. http://dx.doi.org/10.1561/2200000016.

[16] Yang Y, Guan X, Jia Q-S, Yu L, Xu B, Spanos CJ. A survey of ADMM variants for distributed optimization: Problems, algorithms and features. 2022, arXiv:2208.03700.

[17] Engelmann A. Distributed optimization with application to power systems and control (Dissertation), Karlsruhe: Karlsruher Institut für Technologie (KIT); 2021, p. 202. http://dx.doi.org/10.5445/IR/1000127849/v2.
[18] Sun K, Sun XA. A two-level distributed algorithm for nonconvex constrained optimization. Comput Optim Appl 2022. http://dx.doi.org/10.1007/s10589-022-00433-4.
[19] Tang W, Daoutidis P. Fast and stable nonconvex constrained distributed optimization: the ELLADA algorithm. Optim Eng 2021. http://dx.doi.org/10.1007/s11081-020-09585-w, URL: http://link.springer.com/10.1007/s11081-020-09585-w.
[20] Houska B, Frasch J, Diehl M. An augmented Lagrangian based algorithm for distributed nonconvex optimization. SIAM J Optim 2016;26(2):1101–27. http://dx.doi.org/10.1137/140975991, URL: http://epubs.siam.org/doi/10.1137/140975991.
[21] Jiang Y, Sauerteig P, Houska B, Worthmann K. Distributed optimization using ALADIN for MPC in smart grids. 2020, arXiv:2004.01522 [cs, eess, math], URL: http://arxiv.org/abs/2004.01522, arXiv:2004.01522.
[22] Su J, Jiang Y, Bitlislioğlu A, Jones CN, Houska B. Distributed multi-building coordination for demand response. IFAC-Pap 2020;53(2):17113–8. http://dx.doi.org/10.1016/j.ifacol.2020.12.1655.
[23] Engelmann A, Jiang Y, Houska B, Faulwasser T. Decomposition of nonconvex optimization via bi-level distributed ALADIN. IEEE Trans Control Netw Syst 2020;7(4):1848–58. http://dx.doi.org/10.1109/TCNS.2020.3005079.
[24] Rawlings JB, Mayne DQ, Diehl MM. Model predictive control: Theory, computation, and design. 2nd ed.. Madison, Wisconsin: Nob Hill Publishing; 2017.
[25] Stomberg G, Engelmann A, Diehl M, Faulwasser T. Decentralized real-time iterations for distributed NMPC. 2024, arXiv:2401.14898, URL: https://arxiv.org/abs/2401.14898.
[26] Savorgnan C, Romani C, Kozma A, Diehl M. Multiple shooting for distributed systems with applications in hydro electricity production. J Process Control 2011;21(5):738–45. http://dx.doi.org/10.1016/j.jprocont.2011.01.011.
[27] Serale G, Fiorentini M, Capozzoli A, Bernardini D, Bemporad A. Model predictive control (MPC) for enhancing building and HVAC system energy efficiency: Problem formulation, applications and opportunities. Energies 2018;11(3):631. http://dx.doi.org/10.3390/en11030631, URL: http://www.mdpi.com/1996-1073/11/3/631, Number: 3.
[28] Arroyo J, Spiessens F, Helsen L. Comparison of model complexities in optimal control tested in a real thermally activated building system. Build 2022;12(5):539. http://dx.doi.org/10.3390/buildings12050539.
[29] Mork M, Redder F, Xhonneux A, Müller D. Real-world implementation and evaluation of a model predictive control framework in an office space. J Build Eng 2023;78:107619. http://dx.doi.org/10.1016/j.jobe.2023.107619.
[30] Jorissen F, Picard D, Six K, Helsen L. Detailed white-box non-linear model predictive control for scalable building HVAC control. In: 14th modelica conference 2021. 2021, p. 315–23. http://dx.doi.org/10.3384/ecp21181315.
[31] Wei-Han C, Fengqi Y. An energy optimization model for commercial buildings with renewable energy systems via nonlinear MPC. Chem Eng Trans 2022;94:505–10. http://dx.doi.org/10.3303/CET2294084.
[32] Wang C, Wang B, You F. Demand response for residential buildings using hierarchical nonlinear model predictive control for plug-and-play. Appl Energy 2024;369:123581. http://dx.doi.org/10.1016/j.apenergy.2024.123581.
[33] Nelles O. Nonlinear system identification: From classical approaches to neural networks, fuzzy models, and gaussian processes. Cham: Springer International Publishing; 2020, http://dx.doi.org/10.1007/978-3-030-47439-3.
[34] Stoffel P. Learning strategies for data-driven model predictive control of building energy systems (Dissertation), Aachen: RWTH Aachen University; 2024, http://dx.doi.org/10.18154/RWTH-2024-08179, URL: https://publications.rwth-aachen.de/record/992360.
[35] Eser S, Stoffel P, Kumpel A, Muller D. Distributed model predictive control of a nonlinear building energy system using consensus ADMM. In: 2022 30th mediterranean conference on control and automation. MED, Vouliagmeni, Greece: IEEE; 2022, p. 902–7. http://dx.doi.org/10.1109/MED54222.2022.9837140, URL: https://ieeexplore.ieee.org/document/9837140/.
[36] Rätz M, Henkel P, Stoffel P, Streblow R, Müller D. Identifying the validity domain of machine learning models in building energy systems. Energy AI 2024;15:100324. http://dx.doi.org/10.1016/j.egyai.2023.100324.
[37] Stoffel P, Henkel P, Rätz M, Kümpel A, Müller D. Safe operation of online learning data driven model predictive control of building energy systems. Energy AI 2023;14:100296. http://dx.doi.org/10.1016/j.egyai.2023.100296.
[38] Hindmarsh AC, Brown PN, Grant KE, Lee SL, Serban R, Shumaker DE, Woodward CS. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. ACM Trans Math Softw (TOMS) 2005;31(3):363–96. http://dx.doi.org/10.1145/1089014.1089020.
[39] Engelmann A, Jiang Y, Benner H, Ou R, Houska B, Faulwasser T. ALADIN-α – An open-source MATLAB toolbox for distributed non-convex optimization. 2020, arXiv:2006.01866 [cs, eess, math], URL: http://arxiv.org/abs/2006.01866, arXiv:2006.01866.
[40] Biewald L. Experiment tracking with weights and biases. 2020, URL: https://www.wandb.com/, Software available from wandb.com.
[41] Chollet F, et al. Keras. 2015, https://keras.io.