

Article

Modifier Adaptation with Quadratic Approximation with Distributed Estimations of the Modifiers Applied to the MDI-Production Process [†]

Jens Ehlhardt ^{1,*} , Inga Wolf ² and Sebastian Engell ¹ ¹ Process Dynamics and Operations Group, TU Dortmund, Emil-Figge-Straße 70, 42277 Dortmund, Germany² Covestro Deutschland AG, Kaiser-Wilhelm-Allee 60, 51373 Leverkusen, Germany

* Correspondence: jens.ehlhardt@tu-dortmund.de

[†] This paper is an extended version of the “Ehlhardt, J.; Wolf, I.; Engell, S. Real-time optimization with machine learning models and distributed modifier adaptation applied to the MDI-process.” In Proceedings of the 34th European Symposium on Computer Aided Process Engineering and the 15th International Symposium on Process Systems Engineering (ESCAPE34-PSE24), Florence, Italy, 2–6 June 2024.

Abstract

The energy and resource efficient operation of continuously operated large-scale chemical plants is an important factor in the transition towards a sustainable and green process industry. In this work, the operation of the heat exchangers in the diphenylmethane diisocyanate (MDI) process is optimized to reduce fouling and thereby increase their energy efficiency. Real-time optimization (RTO) using Modifier Adaptation With Quadratic Approximation (MAWQA) is applied to cope with plant–model mismatch. It is combined with distributed estimation of the modifiers while retaining a centralized optimization to ensure rapid convergence. It reduces the data points needed for their computation and enables application to large-scale processes. The plant model that is used in the optimization is a surrogate of an available detailed flow-sheet simulator model. The algorithm is demonstrated first for a small problem and then applied to the operator training simulator (OTS) of the MDI process in several operation scenarios. Compared to previous work, the algorithm converges to the optimal operating conditions in fewer iterations.

Keywords: real-time optimization; Modifier Adaptation; Modifier Adaptation with quadratic approximations; isocyanate production; surrogate models; distributed estimation of modifiers



Academic Editors: Yuanqiang Zhou, Le Yao, Xiaoyu Jiang and Zheren Zhu

Received: 31 August 2025

Revised: 23 September 2025

Accepted: 23 September 2025

Published: 30 September 2025

Citation: Ehlhardt, J.; Wolf, I.; Engell, S. Modifier Adaptation with Quadratic Approximation with Distributed Estimations of the Modifiers Applied to the MDI-Production Process. *Processes* **2025**, *13*, 3140. <https://doi.org/10.3390/pr13103140>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Today, the chemical industry faces a huge challenge: the transition to reduced CO₂ footprints and enhanced sustainability while staying competitive. Optimizing the operation of plants and processes that cannot be directly replaced by sustainable alternatives is an important factor in reducing the industry’s emissions. One example of large-scale energy intensive processes is the production of polyurethanes and their precursors, which are used, e.g., in the production of foams. These foams are used in applications ranging from shoes and mattresses to refrigerators, building insulation, and cars [1].

The most widely used precursor of polyurethanes is diphenylmethane diisocyanate (MDI). In the current production process, large heat inputs are needed, which are mainly provided in the form of steam and transferred to the process via heat exchangers. Fouling in the heat exchangers reduces their efficiency and eventually leads to the need of cleaning and plant shutdowns. The rate of fouling depends on the operating conditions of the

process. Consequently, there is a strong interest to optimize the operation of the heat exchangers to reduce the fouling while maintaining high productivity levels [2].

This task is within the scope of real-time optimization (RTO), where numerical optimization is typically employed at relatively long sampling times to optimize the stationary operating conditions of, e.g., a chemical plant. For RTO, a mathematical model of the process is needed that accurately describes the plant and can be coupled with modern optimization solvers [3]. For many large-scale production processes, detailed steady-state simulation models are available in designated software: so-called flow-sheet simulators. These are either commercial products or in-house tools of the producing companies. Such simulators are designed for the interactive use by process designers and the emphasis is on very detailed and accurate descriptions of the behaviour of the processes, at the expense of computational efficiency. Moreover, convergence problems may occur and often, there is no option offered to couple them directly to real-time data acquisition and process control systems. In [4], employing artificial neural networks (ANNs) as surrogate models was proposed, which approximate the detailed flow-sheet simulation of complex processes, as well as using these surrogate models in an RTO scheme. Thus, the flow-sheet simulator acts as an offline data source or digital twin for the identification of the ANN models, which are then employed in the online application of RTO. The combination of surrogate modeling and digital twins enables the fast execution of the underlying models as well as the integration of measured data or only partly understood phenomena into optimization [5]. The idea is to use surrogate models of flow-sheet simulators for large-scale optimization [6].

Generally, RTO schemes need to handle the inevitable mismatch between the optimization model and the real behaviour of the plant; due to this discrepancy, the application of the optimal set-points that were computed for the plant model may not lead to an optimal operation of the real plant. This can result in sub-optimal performance or even infeasible operating conditions [7]. Several approaches have been proposed to address the issue of plant-model mismatch. The best known and most widely used approach is the two-step approach, see, e.g., [8], where, at each sampling time or at longer intervals, some parameters of the plant model are updated based on the available measurements and the optimization is performed using the updated model. The number of parameters that can be estimated depends on the available information. The two-step approach is tailored to the situation where the model is structurally correct so that it can represent the true plant behaviour for a suitable set of parameters. However, strictly speaking, this is never true, and practically, there may be significant unmodelled effects and phenomena. In this case, parameter updates may even lead to worse results than using a nominal model [9] and in general, the true optimum may not be achieved. Nonetheless, the two-step approach is the standard RTO methodology in industry, especially in the petroleum industry [10]. Often, it is applied in combination with (linear) model predictive control. Therefore, the economics are optimized via RTO and constraint satisfaction is enforced by a model predictive controller [11].

In Modifier Adaptation (MA), instead of adapting model parameters, the optimization problem itself is updated. Zeroth-order (bias) and first-order (gradient) correction terms, also called modifiers, are added to the cost and constraint functions to match the Karush-Kuhn-Tucker conditions (KKT) and to achieve the true plant optimum up on convergence. This was first proposed by Tatjewski [12] and extended to problems with constraints in [13]. While bias correction is well-known and easy to implement, the estimation of the gradients of the cost function and of the constraints of the real plant is the most challenging element of the MA approach. They can be approximated by finite difference (FD) methods, as discussed in [13,14], making use of available measurements rather than perturbing the plant at each iteration possible. However, using finite differences leads

to a high sensitivity to measurement noise or disturbances for small step sizes and to inaccurate approximations for larger step sizes. To cope with this problem, in [15], a new technique, called Modifier Adaptation With Quadratic Approximations (MAWQA), was proposed where the cost and constraint functions of the plant are approximated by quadratic functions. The required gradients are then calculated based on the quadratic approximations. Due to the smoothing effect of the least-squares minimization in the identification of the approximations, MAWQA is more robust to noise. MAWQA was inspired by “derivative-free optimization” as proposed in [16] and incorporates techniques for data-point selection [17], the use of trust regions, and a model quality check. MAWQA has been successfully implemented to lab- and pilot-scale plants, as shown in [18–21]. A combination of MAWQA and Effective Model Adaptation (EMA) was successfully demonstrated at a lab-scale membrane plant. In EMA, the model parameters are updated while enforcing its adequacy for the optimization [22].

A drawback of MAWQA is that the minimum number of data points that are needed to compute the quadratic approximations is a quadratic function of the number of inputs. This limits its application to large-scale plants, where the efficient quadratic approximations can only be used for the first time after a relatively large number of probing steps and iterations have been performed, e.g., 21 plant evaluations for five inputs that are optimized. This problem can be addressed by combining Modifier Adaptation with distributed optimization. The goal of the application of decomposition techniques is to split the optimization problem into a set of smaller problems, each with fewer inputs. In [23], Modifier Adaptation was investigated in the context of distributed optimization. Three different algorithms were presented, which exchange different amounts of information between the subsystems. In all three algorithms, the modifiers are identified globally with respect to all decision variables. The focus of the study was on assuring privacy between the different subsystem operators (e.g., different plants in a chemical site), which is paid for by reducing the speed of convergence. In [24], a similar approach was successfully applied to gas compressor stations both in simulations and lab-scale settings. The modifiers are estimated locally on a subsystem-level and finite differences are used for the gradient estimation. In [25], the same approach was applied to output Modifier Adaptation (MAy). In MAy, the model outputs rather than the objective and constraint functions are adapted. Consequently, the modifiers of the sub-models can be computed individually and the modified outputs are then fed into the global constraint and objective functions.

Another approach was presented by [26]. Here, distributed feedback optimizing control is used to apply RTO to a gas-lifted oil well rig in simulation. The RTO problem is stated as a cascaded control structure that resembles the dual descent method. In the inner loop, the gradients of the Lagrangian are controlled to be zero, while in the slower outer loop, the Lagrange multipliers are updated to satisfy the constraints. However the poor scalability in combination with the complex control structure design for larger systems limits its application [27]. In [28], the approach was applied to a lab-scale gas-lifted oil rig.

Ref. [29] proposed using Gaussian processes in Modifier Adaptation. This was demonstrated in the optimization of wind farms [30] and for a lab-scale gas-lifted oil-rig [31].

Ref. [32] proposed another method to reduce the number of gradients that are estimated with the goal of enabling the application of MA to larger systems. The plant gradients are only computed with respect to few privileged directions in the inputs space, which are identified based on the sensitivity of the Lagrangian with respect to some uncertain parameters. The scheme was successfully applied to optimize the trajectory of an airborne energy system. It has not yet been investigated though as to how this could be generalized to structural plant–model mismatch. Despite all these advances in Modifier Adaptation and beyond, it still lacks acceptance and application in the industry [10].

In this work, MAWQA is combined with distributed estimation of the required modifiers to enable its application to large-scale systems. The problem is decomposed by means of introducing coupling variables while retaining a centralized optimization to ensure rapid convergence. Moreover, it avoids the convergence problems of distributed optimization algorithms (see, e.g., [33]). In this way, decomposition is performed only for the identification of the local quadratic approximations for the estimation of local modifiers. In addition, the decision between solving the modified optimization problem and an optimization problem based on the quadratic approximation, the trust-region constraints, and potentially required additional perturbations are handled on the subsystem level. The distribution of the modifier estimation problem thereby exploits the structure of the plant at hand. Hence, the required number of perturbations can be reduced and the speed of convergence to the true optimum of the overall plant can be improved. If the decomposition is conducted with respect to subunits of a plant, there is usually no need to respect limitations of the sharing of data; hence, an integrated optimization is feasible. The method presented in this paper has successfully been applied to the operator training simulator of the MDI-production process of the MDI producer Covestro, which provides an accurate digital twin of the real plant. The model of the OTS is structurally different from the model used in the flowsheet simulator and shows a different behaviour. In comparison to previous work [4], faster convergence is achieved. The distributed MAWQA method was first been proposed in [34]. In this paper, we extend this work and provide a detailed analysis for a small case study as well as for the MDI process.

The remainder of the paper is structured as follows: Section 2 describes the concepts of Modifier Adaptation and MAWQA, as well as the distributed estimation of modifiers. In addition, the distributed scheme is applied to a small case study. Section 3 provides an overview of the investigated MDI process. Sections 4 and 5 present the application of the distributed MAWQA scheme to the MDI process and the results obtained. Finally, Section 6 provides a conclusion to the paper and directions for further research.

2. Real-Time Optimization Using Modifier Adaptation

2.1. Real-Time Optimization

The goal of real-time optimization is to compute the optimal steady-state operating conditions of a plant or process. Mathematically, the objective is to find the solution of the optimization problem ((1) and (2)).

$$\min_{\mathbf{u}} J_p(\mathbf{u}) \quad (1)$$

$$s.t. \mathbf{G}_p(\mathbf{u}) \leq \mathbf{0} \quad (2)$$

Throughout this work, vectors and matrices are written in bold. J_p is the cost function of the plant, $\mathbf{u}$ is a vector of inputs that can be influenced, and $\mathbf{G}_p$ represents a set of constraint functions. The plant model is contained in $\mathbf{G}_p$ as an equality constraint. However, the true plant cost function and constraints, indicated by the subscript p , are generally not known due to plant-model mismatch and unknown influences. Therefore, ((1) and (2)) is replaced by the model-based optimization.

$$\min_{\mathbf{u}} J_M(\mathbf{u}) \quad (3)$$

$$s.t. \mathbf{G}_M(\mathbf{u}) \leq \mathbf{0} \quad (4)$$

In (3) and (4), the assumed cost and constraint functions (including the nominal model) are indicated by the subscript M . Due to plant-model mismatch and unknown influences, the

optimal solution of the optimization problem for the plant, $\mathbf{u}_P^*$, and the optimal solution of the model, $\mathbf{u}_M^*$, do not match,

$$\mathbf{u}_M^* \neq \mathbf{u}_P^*, \quad (5)$$

and the constraints of the true plant may not be met at $\mathbf{u}_M^*$.

2.2. Modifier Adaptation with Quadratic Approximation

In Modifier Adaptation (MA) or iterative gradient correction, the model-based optimization problem ((3) and (4)) is extended by affine correction terms in the objective and constraint functions to match the conditions of optimality of the real plant [13]:

$$\min_{\mathbf{u}^{k+1}} J_{ad}^k(\mathbf{u}^{k+1}) := J_m(\mathbf{u}^{k+1}) + \lambda_J^k(\mathbf{u}^{k+1} - \mathbf{u}^k) \quad (6)$$

$$\text{s.t. } \mathbf{G}_{ad}^k(\mathbf{u}^{k+1}) := \mathbf{G}_m(\mathbf{u}^{k+1}) + \lambda_G^k(\mathbf{u}^{k+1} - \mathbf{u}^k) + \epsilon_G^k \leq \mathbf{0} \quad (7)$$

$$\mathbf{u}^{lb} \leq \mathbf{u}^{k+1} \leq \mathbf{u}^{ub}. \quad (8)$$

In every iteration k , the adapted optimization problem ((9)–(11)) is solved to find the next vector of operating conditions $\mathbf{u}^{k+1}$. $\mathbf{u}^{lb}$ and $\mathbf{u}^{ub}$ are the lower and upper bounds of the degrees of freedom. The affine correction factors, so-called modifiers, are added to correct the gradient of the cost function and the absolute values and the gradients of the constraint functions [15]. The bias modifier ϵ_G and the gradient modifiers λ_G and λ_J at iteration k are defined as

$$\epsilon_G^k = \mathbf{G}_P^k - \mathbf{G}_M^k \quad (9)$$

$$\lambda_J^k = \nabla_{\mathbf{u}} J_P^k - \nabla_{\mathbf{u}} J_M^k \quad (10)$$

$$\lambda_G^k = \nabla_{\mathbf{u}} \mathbf{G}_P^k - \nabla_{\mathbf{u}} \mathbf{G}_M^k. \quad (11)$$

$\nabla_{\mathbf{u}}$ denotes the gradients of the cost or constraint functions with respect to the inputs. As the plant functions are not known, the gradients need to be determined based on measured data. Often, finite differences are used [14]. However, the computation via FD is sensitive to noise and the selection of an appropriate step size is challenging. Gao et al. [15] proposed identifying the process gradients based on quadratic approximations of the cost and constraint functions and called the resulting scheme Modifier Adaptation with Quadratic Approximation (MAWQA). The quadratic approximations are of the form (12)

$$J_{\Phi}(\mathbf{u}, \boldsymbol{\theta}) = \sum_{i=1}^{N_u} \sum_{j=1}^i a_{i,j} u_i u_j + \sum_{i=1}^{N_u} b_i u_i + c \quad (12)$$

$$\boldsymbol{\theta} = [a_{1,1}, \dots, a_{N_u, N_u}, b_1, \dots, b_{N_u}, c]^T. \quad (13)$$

In ((12) and (13)), N_u is the number of inputs and the parameters of the quadratic functions $a_{i,j}$, b_i , and c are subsumed in the parameter vector $\boldsymbol{\theta}$. Φ indicates which function is approximated. Quadratic functions for the constraints are formed analogously. The gradients of the cost and constraint functions can then be robustly approximated according to

$$\nabla_{\mathbf{u}} J_P^k \approx \nabla_{\mathbf{u}} J_{\Phi}^k \quad (14)$$

and

$$\nabla_{\mathbf{u}} \mathbf{G}_P^k \approx \nabla_{\mathbf{u}} \mathbf{G}_{\Phi}^k. \quad (15)$$

The MAWQA workflow is depicted in Figure 1.

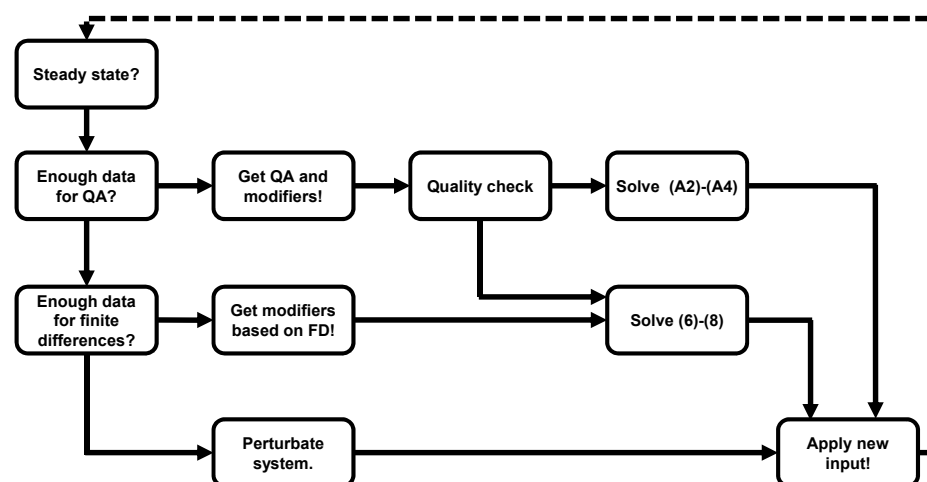


Figure 1. Schematic flow diagram of the Modifier Adaptation with Quadratic Approximation algorithm used for RTO. When a steady state has been reached, it is evaluated whether finite-difference or quadratic approximations can be used for gradient estimation. If too few data points are available, new setpoints are generated by perturbation. In the case of QA, a quality check is performed to decide whether the modified optimization problem or the optimization based on the quadratic approximation should be used. In all cases, the new setpoints are applied to the plant, followed by waiting for the next steady state.

In every iteration, it is checked whether enough and sufficiently rich data are available for the quadratic approximations or gradient computation by finite differences. If neither criterion is fulfilled, the current set-points are perturbed and the plant is probed. If enough data are available for computing finite differences, the gradients are computed by a local linear approximation of the cost or constraint functions; the modifiers are calculated and the modified optimization problem (6)–(8) is solved to obtain the next set-points. If enough data points are available for computing quadratic approximations, a regression set is formed and the quadratic approximations are identified. To ensure a regression set that is well-posed for the identification of the quadratic functions, a subset of all available data points is chosen by an algorithm as presented in [17]. The gradients are then computed analytically based on the quadratic approximations. Dependent on quality measures, see (A5) and (A6), the modified problem (6)–(8) or an optimization problem based on the quadratic approximations is solved. In both cases, a trust region constraint is added to the optimization problems. The new set-points are applied and the algorithm is repeated as soon as a new steady-state has been reached. For more information, the interested reader is referred to in Appendix A and [15].

2.3. Distributed Modifier Adaptation with Quadratic Approximation

A bottleneck of the application of Modifier Adaptation schemes to plants with a significant number of inputs, and especially of MAWQA, is the number of perturbations required for gradient estimation. Their number grows linearly for finite differences and quadratically for quadratic approximations. A comparison is shown in Figure 2. Even for a medium number of inputs (≥ 4), MAWQA is difficult to apply in reality due to the required large number of perturbations. Therefore, exploiting the decomposable structure of the plants to extend the application of MAWQA to large-scale plants is proposed. Note that we do not necessarily decompose the solution of the optimization problem, as we assume the real-time optimization of a complex plant under single ownership and feasibility of the solution of the problem within the available computation time, but only the estimation of

the modifiers. This ensures a fast convergence and avoids the convergence problems of fully distributed algorithms.

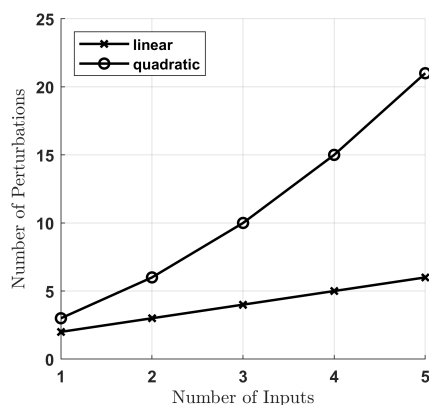


Figure 2. Minimum number of perturbations needed for gradient estimation using finite differences and quadratic approximations. The number of perturbations needed for the QA increases quadratically. Even for a system with four or more inputs, QA becomes difficult to apply due to the number of initial data points required.

Many processes and plants consist of a network of different unit operations, which, however, need to be optimized simultaneously; examples include chemical production sites, heat exchanger networks, and wind farms. We assume that the problem at hand can be represented by the general formulation of a decomposable optimization problem with an additive cost function and local as well as plant-wide constraints as given in (16)–(18):

$$\min_{\mathbf{u}_1, \dots, \mathbf{u}_N} J_{tot} = \sum_{i=1}^N J_i(\mathbf{u}_i) \quad (16)$$

$$s.t. \sum_{i=1}^N \mathbf{G}_{ov,i}(\mathbf{u}_i) - \mathbf{G}_{ov}^{tot} \leq \mathbf{0} \quad (17)$$

$$\mathbf{G}_{loc,i}(\mathbf{u}_i) \leq \mathbf{0}, i = 1, \dots, N. \quad (18)$$

The overall cost function J_{tot} is assumed to be the summation of the cost functions J_i of the N subsystems, which only depend on the individual inputs $\mathbf{u}_i$. These may, e.g., be the operating cost or the energy consumption of the different units. In addition, the subsystems possess local constraints $\mathbf{G}_{loc,i}$ and are coupled via Equation (17), also called the complicating constraint. These constraints can, for example, express shared resources (steam, electricity, etc.) [33]. Auxiliary variables (19) can be introduced to decompose optimization problems that are initially not decomposable, as follows:

$$\mathbf{v}_i = \mathbf{h}(\mathbf{u}_1, \dots, \mathbf{u}_j, \dots, \mathbf{u}_N), i \neq j \quad (19)$$

$\mathbf{v}_i$ are coupling or interconnection variables and $\mathbf{h}(\bullet)$ is assumed to be a known, differentiable function. Such variables may, e.g., represent streams between process units. The connection variables are assumed to be fully defined based on the inputs of the other subsystems. Combining the standard MA equations ((6)–(8)), the coupling variables (19), and distributed optimization problem ((16)–(18)), a Modifier Adaptation problem can be

formulated. The decomposable structure can then be exploited to identify the required modifiers on the subsystem level.

$$\min_{\mathbf{u}^{k+1}, \mathbf{v}^{k+1}} J_{ov,ad}^k = \sum_{i=1}^N J_{ad,ov,i}^k(\mathbf{u}_i^{k+1}, \mathbf{v}_i^{k+1}) \quad (20)$$

$$s.t. \mathbf{G}_{ov,ad}^k = \sum_{i=1}^N \mathbf{G}_{ov,ad,i}^k(\mathbf{u}_i^{k+1}, \mathbf{v}_i^{k+1}) - \mathbf{G}_{ov}^{tot} \leq \mathbf{0} \quad (21)$$

$$\mathbf{v}_i^{k+1} = \mathbf{h}(\mathbf{u}_1^{k+1}, \dots, \mathbf{u}_j^{k+1}, \mathbf{u}_N^{k+1}), \forall i \in N, j \neq i \quad (22)$$

$$\mathbf{G}_{loc,ad,i}^k(\mathbf{u}_i^{k+1}, \mathbf{v}_i^{k+1}) \leq \mathbf{0}, i = 1, \dots, N \quad (23)$$

$$\mathbf{u}_i^{lb} \leq \mathbf{u}_i^{k+1} \leq \mathbf{u}_i^{ub}, i = 1, \dots, N \quad (24)$$

The separable overall adapted cost function $J_{ad,ov}$ is the summation of the adapted cost functions of the individual N sub-models $J_{ad,ov,i}$. Analogously, the adapted complicating constraint functions $\mathbf{G}_{ov,ad}$ are the sum of the adapted contributions of the sub-models $\mathbf{G}_{ov,ad,i}$. $\mathbf{u}_i$ and $\mathbf{v}_i$ are the set-points and coupling variables of the i -th subsystems. $\mathbf{G}_{ad,loc,i}$ are local constraints and $\mathbf{u}_i^{lb}$ and $\mathbf{u}_i^{ub}$ represent the bounds of the inputs of the i -th subsystem.

$$J_{ov,ad,i}^k := J_{ov,m,i}(\mathbf{u}_i^{k+1}, \mathbf{v}_i^{k+1}) + \lambda_{J,i}^k(\mathbf{z}_i^{k+1} - \mathbf{z}_i^k) \quad (25)$$

$$\mathbf{G}_{ov,ad,i}^k := \mathbf{G}_{ov,m,i}(\mathbf{u}_i^{k+1}, \mathbf{v}_i^{k+1}) + \lambda_{\mathbf{G},ov,i}^k(\mathbf{z}_i^{k+1} - \mathbf{z}_i^k) + \epsilon_{\mathbf{G},ov,i} \quad (26)$$

$$\mathbf{G}_{loc,ad,i}^k := \mathbf{G}_{loc,m,i}(\mathbf{u}_i^{k+1}, \mathbf{v}_i^{k+1}) + \lambda_{\mathbf{G},loc,i}^k(\mathbf{z}_i^{k+1} - \mathbf{z}_i^k) + \epsilon_{\mathbf{G},loc,i} \quad (27)$$

Similar to (6) and (8), the adapted cost and constraint functions are summations of the model functions and affine correction terms. In the distributed case, the modifiers are defined in terms of the variables $\mathbf{z}_i$ instead of the original full input vector $\mathbf{u}$, according to

$$\mathbf{z}_i = [\mathbf{u}_i^T, \mathbf{v}_i^T]^T \quad (28)$$

$$\lambda_{J,ov,i}^k = \nabla_{\mathbf{z}_i} J_{P,ov,i}^k - \nabla_{\mathbf{z}_i} J_{M,ov,i}^k \quad (29)$$

$$\lambda_{\mathbf{G},ov,i}^k = \nabla_{\mathbf{z}_i} \mathbf{G}_{P,ov,i}^k - \nabla_{\mathbf{z}_i} \mathbf{G}_{M,ov,i}^k \quad (30)$$

$$\lambda_{\mathbf{G},loc,i}^k = \nabla_{\mathbf{z}_i} \mathbf{G}_{P,loc,i}^k - \nabla_{\mathbf{z}_i} \mathbf{G}_{M,loc,i}^k \quad (31)$$

$$\epsilon_{\mathbf{G},ov,i}^k = \mathbf{G}_{P,ov,i}^k - \mathbf{G}_{M,ov,i}^k \quad (32)$$

$$\epsilon_{\mathbf{G},loc,i}^k = \mathbf{G}_{P,loc,i}^k - \mathbf{G}_{M,loc,i}^k \quad (33)$$

$\mathbf{z}_i$ represents the set-points and the coupling variables of the individual subsystems. The maximum dimension of $\mathbf{z}_i$ is often smaller than the dimension of the overall input vector $\mathbf{u}$, see Equation (34). The gradients $\nabla_{\mathbf{z}_i}$ are calculated with respect to the inputs $\mathbf{z}_i$ and the quadratic functions (12) are based on the variables $\mathbf{z}_i$ instead of $\mathbf{u}$. If

$$N_{\mathbf{z},max} = \max(\dim(\{\mathbf{z}_1, \dots, \mathbf{z}_N\})) < \dim(\mathbf{u}), \quad (34)$$

less data points are needed for the calculation of the gradients related to all subsystems and, thus, for the identification of the quadratic functions. A limitation here is that it must be possible to compute the coupling variables from the applied inputs or they must be measurable and exhibit sufficient variation.

The flow-chart of the resulting algorithm is shown in Figure 3. First, $N_{\mathbf{z},max} + 1$ initial perturbations are applied to the plant. Then, enough data points are available such that the gradients can be estimated via the method presented in [13]. The next steps are carried out for each sub-model individually. Gradients and/or quadratic functions are calculated and the decision between choosing the adapted optimization problem or the problem based

on the quadratic function ((A2)–(A4)) is taken for each sub-model. Afterwards, all local modified cost and constraint functions are assembled to form the overall optimization problem. Problem ((20)–(24)) is solved and the new input is applied to the plant. Convergence is checked and the algorithm either is terminated or repeated. The optimization problem Equations (20)–(24) can be solved in a centralized fashion or with distributed approaches, e.g., alternating direction method of multipliers (ADMM) [35].

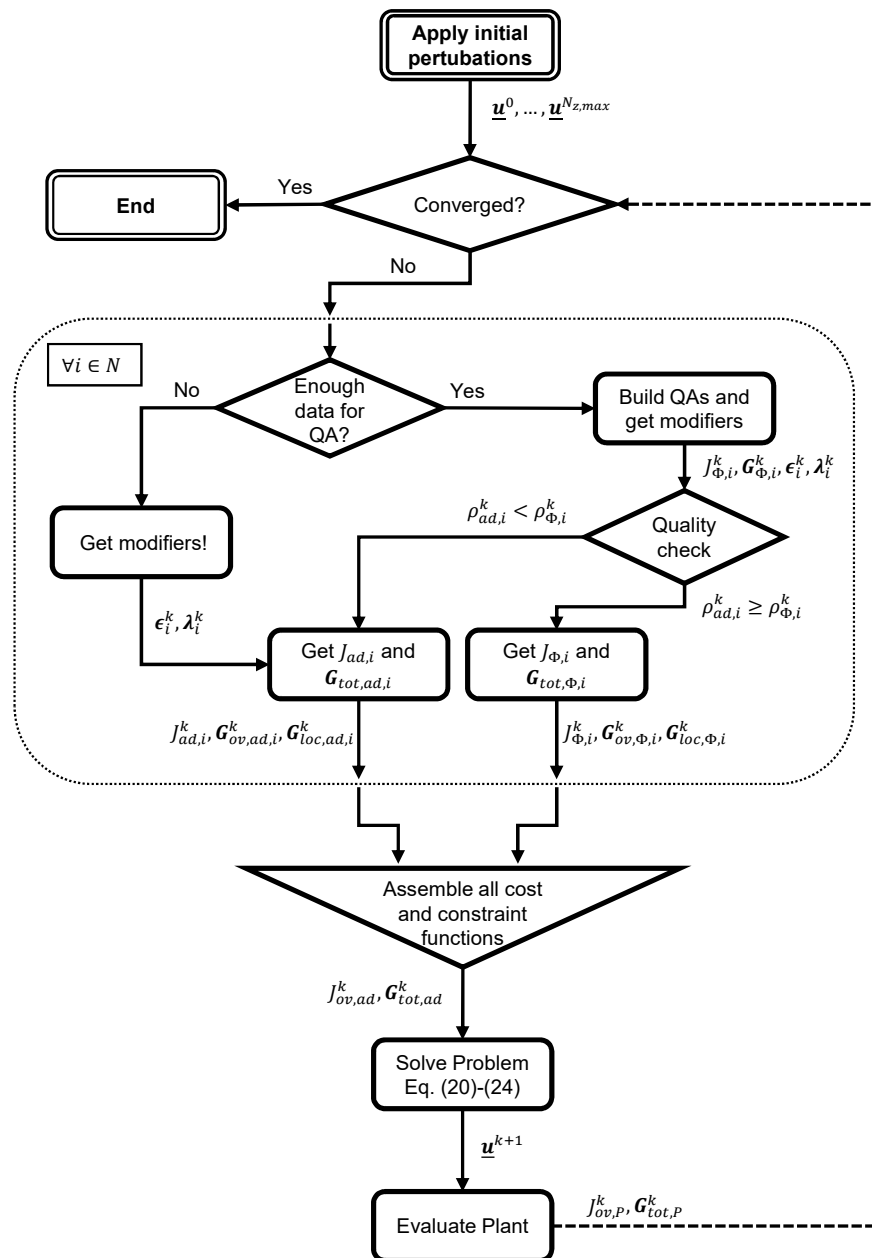


Figure 3. Schematic flow diagram of the distributed MAWQA scheme. If no data points are available, initial perturbations are applied to the system. In each iteration of the disMAWQA algorithm, convergence is checked. If the user-defined convergence criterion is not met, the choice between finite differences and quadratic approximations, as well as the quality check, is performed for each subsystem individually. The overall centralized optimization problem is then assembled from all individual objective and constraint functions. This problem is solved, and the new setpoints are applied to the system. Note that the overall problem may include modified objective and constraint functions in addition to quadratic approximations.

2.4. Demonstration for a Small Example

In this Section, the methodology presented in Section 2.3 is applied to a small system for demonstration purposes. In the first scenario, the number of subunits N is fixed to 3 to demonstrate the convergence of the approach. In the second scenario, the number of subunits is increased, to showcase the advantage of MAWQA with distributed estimation of the modifiers (disMAWQA) when applied to larger systems. The structure of the plant is displayed in Figure 4. In the first stage, units are operated in parallel with their associated inputs u_1 to u_{N-1} . The outgoing streams are mixed and fed into unit N . The mixing can be described by a mixing rule according to

$$v = \frac{1}{N-1} \sum_{i=1}^{N-1} u_i. \quad (35)$$

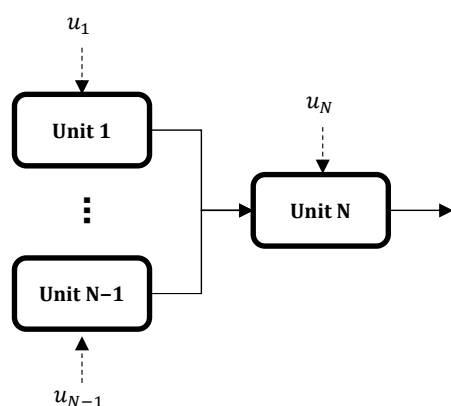


Figure 4. General structure of the minimal example.

The optimization problem is given in (36)–(38).

$$\min_{u_1, \dots, u_N, v} \sum_{i=1}^N J_i \quad (36)$$

$$\text{s.t. } u_i^{lb} \leq u_i \leq u_i^{ub}, i \in \{1, 2, \dots, N\} \quad (37)$$

$$v = \frac{1}{N-1} \sum_{i=1}^{N-1} u_i \quad (38)$$

with

$$J_i = \begin{cases} a_i(u_i - b_i)^2, & i \in \{1, 2, \dots, N-1\} \\ a_i(u_i - b_i - v)^2, & i = N \end{cases} \quad (39)$$

$$a_i = \begin{cases} \{2.4 + i \times \frac{0.7}{N-1}\}, & i \in \{1, 2, \dots, N-1\} \\ 3.1, & i = N \end{cases} \quad (40)$$

$$b_i = \begin{cases} \{1.5 + i \times \frac{2}{N-1}\}, & i \in \{1, 2, \dots, N-1\} \\ 4, & i = N. \end{cases} \quad (41)$$

The overall cost function is the summation of the individual cost functions J_1 to J_N . Equation (37) bounds the inputs and the interconnection variable v is added as a constraint (38). The parameters a_i and b_i are defined according to Equations (40) and (41). The plant model and the optimization model have the same structure but different values of b_i . In this example, no complicating or local constraints are present and no noise is added to the problem. The offset in the parameters is given by

$$b_i = \begin{cases} \{1 + i \times \frac{2}{N-1}\}, & i \in \{0, 1, \dots, N-1\} \\ 5, & i = N. \end{cases} \quad (42)$$

In the first scenario, the number of subunits is 3 and the total number of inputs is $N_u = 3$. Accordingly, four perturbations are needed for computing finite differences and $(3+1)(3+2)/2 = 10$ perturbations for the quadratic approximations. With the distributed estimation of the modifiers, the number of perturbations can be reduced. The number of necessary perturbations and the necessary gradients are listed in Table 1. Quadratic approximations for all models are available after six perturbations.

Table 1. Properties of the small system with $N = 3$.

	Unit 1	Unit 2	Unit 3
$\mathbf{z}_i$	$[u_1]$	$[u_2]$	$[u_3, v]^T$
$N_{min,FD}$	2	2	3
$N_{min,QA}$	3	3	6
$\nabla_{\mathbf{z}_i} J_i$	$[\frac{\partial J_1}{\partial u_1}]$	$[\frac{\partial J_2}{\partial u_2}]$	$[\frac{\partial J_3}{\partial u_3}, \frac{\partial J_3}{\partial v}]^T$

The resulting trajectories of the first scenario are shown in Figure 5. It can be seen that the distributed modifier estimation approach converges to the optimal values of the inputs in iteration 12 while the standard approach needs 15 iterations. The same settings were used for both approaches. All optimization problems were solved in Matlab (R2022a) [36] using the OPTI toolbox (v2.29) [37].

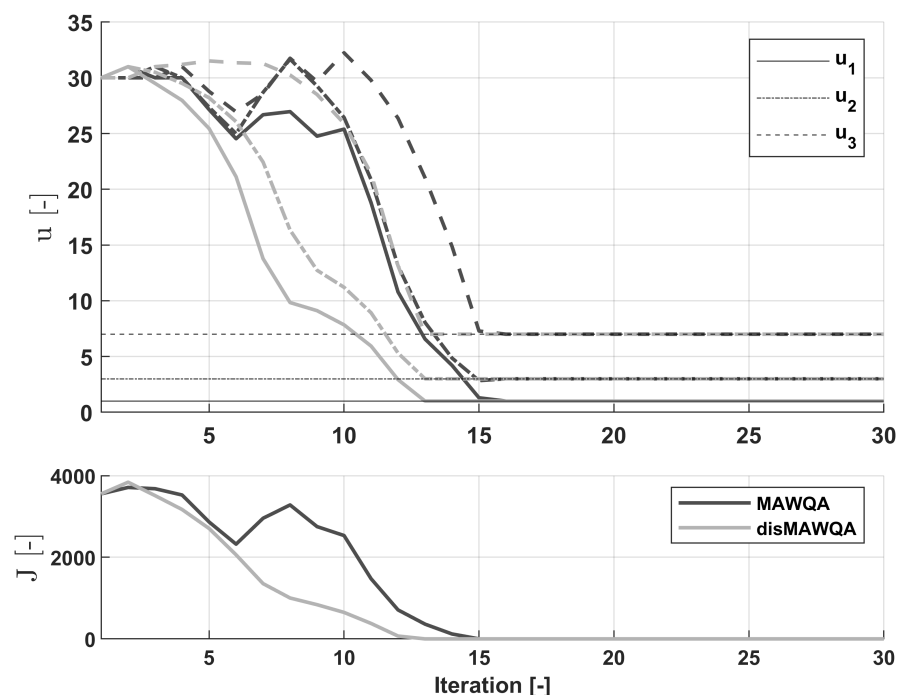


Figure 5. Trajectories of the inputs (upper plot) and objective function values (lower plot) of MAWQA and MAWQA with the distributed estimation of modifiers (disMAWQA) for the demonstration example with three subsystems. On the x-axes, the iteration number is displayed.

In the second scenario, the advantage of the distributed approach when applied to larger systems is investigated. The number of subunits is varied between 3 and 11 and the resulting trajectories are displayed in Figure 6.

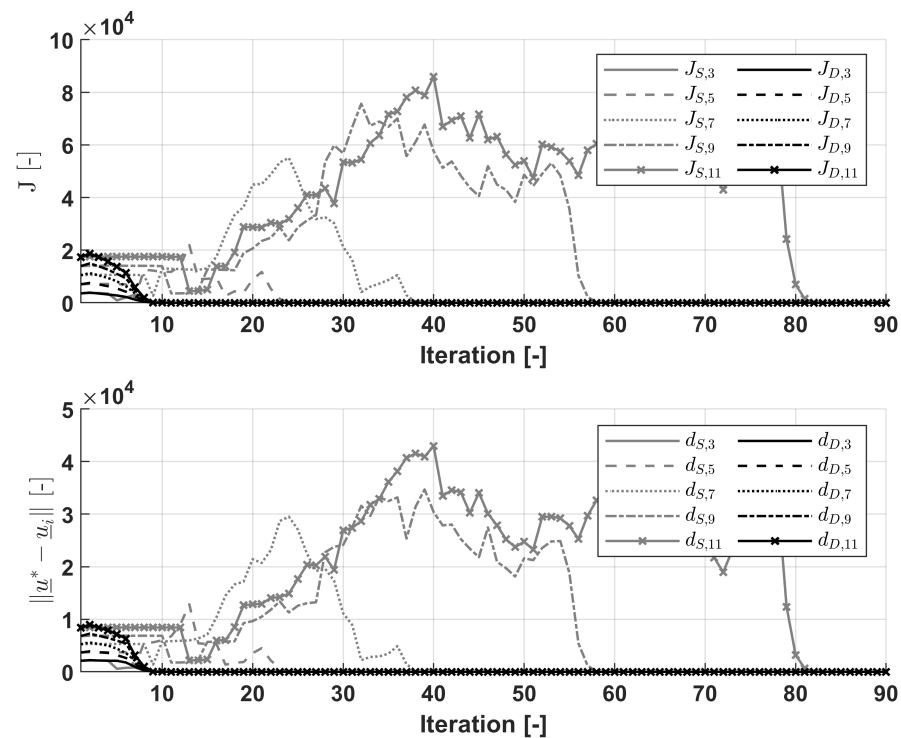


Figure 6. In the upper plot, the trajectories of the objective values of the standard MAWQA $J_{S,N}$ and the disMAWQA approach $J_{D,N}$ are shown. In the lower plot, the Euclidean distance to the optimal input vector u^* of the standard MAWQA approach $d_{S,N}$ and the disMAWQA approach $d_{D,N}$ are presented. On the x-axes, the iteration number is displayed.

In Figure 6, the trajectories for the objective values of the standard MAWQA algorithm $J_{S,N}$ and the disMAWQA algorithm $J_{D,N}$ as well as the Euclidean distances to the optimal operating points of the standard MAWQA algorithm $d_{S,N}$ and the disMAWQA algorithm $d_{D,N}$ are displayed in the upper and lower plot. N represents the number of subunits in the plant, which are highlighted by the consistent line types. In both plots, the iteration number is displayed on the x-axes. As in the first case, the disMAWQA approach converges to the optimal operating point in iteration 12. For disMAWQA, this number stays constant even for larger systems. The iterations needed for convergences in contrast increase drastically for the standard MAWQA algorithm. In the case of 11 subunits, more than 80 iterations are needed. Both approaches converge to the optimal operating points as the Euclidean distance converges to zero, as is visible in the lower subplot. Due to the quadratic approximations, the number of perturbations increase quadratically with the number of inputs of the system. In the disMAWQA approach, the quadratic approximations are identified locally on a subunit level with only 1 or 2 inputs each. The standard MAWQA approach still relies on all inputs of the whole system. The advantage of disMAWQA is caused by the exploitable structure of the system and accordingly, the lower number of perturbations needed for the gradient estimation.

3. Isocyanate Production Process

Here, we consider one process stage of the production of diphenylmethane diisocyanate (MDI), as described in [4]. In the 1930s, the industrial production of MDI became important after the addition polymerization of difunctional isocyanates was discovered by O. Bayer. Methylene diphenyl diisocyanates are important feedstocks for the production of polyurethanes, and diphenylmethane diisocyanate (MDI) is the most widely used precursor of polyurethanes [1]. Polyurethanes are a main component in the foam production with

end-user applications ranging from mattresses and footwear to refrigerators, insulation panels, and cars.

In the process stage that is considered here, methylenedianiline (MDA) reacts to MDI in several reactors. Thereafter, MDI is separated from other gaseous reaction products. During the reaction and the separation step, several heat exchangers transfer heat to the process that is provided in the form of steam [4]. The objective of this work is to improve the energy efficiency of the MDI process and reduce its downtime by optimally distributing the total heat required in this process stage among the available heat exchangers, so that fouling on the process side is minimized and cleaning operations are needed as infrequently as possible [2]. The speed of the fouling processes is directly correlated to the pressure and temperature of the steam. If the steam temperature in a heat exchanger is high, fouling processes are accelerated on the process side of that heat exchanger. In order to be able to transfer the same amount of heat for this state of increased fouling, the steam temperature has to be increased, which results in more accelerated fouling such that cleaning activities are soon unavoidable. Rather than applying the same amount of heat input to the heat exchangers subject to increased fouling, it is preferable to transfer part of the required heat via the heat exchangers subject to less fouling. This leads to the goal that the steam pressures of all heat exchangers remain close to their clean values. In this manner, not only is the overall steam demand reduced but the time until the next cleaning activities are necessary is also extended.

The process consists of two stages. In the reaction stage, MDA reacts to MDI in multiple reactors that are operated in parallel. The outlet streams of the reactors are mixed and further processed in the separation stage, where multiple units are operated in sequence. In each unit, heat is supplied via heat exchangers. Figure 7a shows the scheme of the MDI-plant. Each heat exchanger is equipped with a cascaded control structure, see Figure 7b, that consists of a temperature controller in the outer loop and a pressure controller in the inner loop.

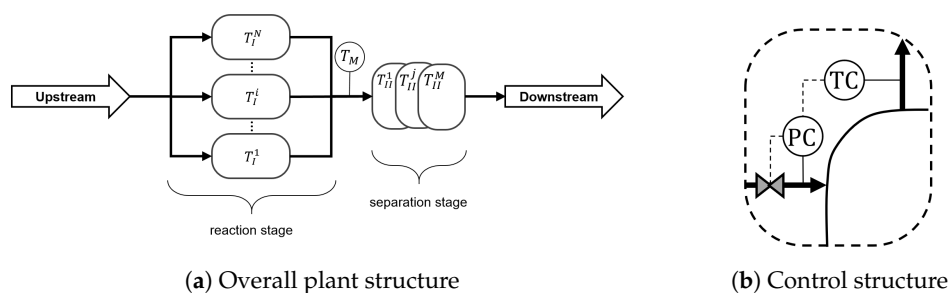


Figure 7. In (a), a schematic representation of the plant is shown. In the reaction stage, multiple units are operating in parallel with the corresponding temperature set-points of the heat exchangers T_i^j . The outlet streams are mixed, and the resulting mixing temperature is denoted as T_M . After mixing, the streams are further processed in the separation stage with the corresponding temperature set-points T_{II}^j . The employed control structure is shown in (b).

3.1. Surrogate Modeling of the MDI Process

In order to implement a model-based RTO scheme, a mathematical model that represents the process under consideration is needed. A physics- and chemistry-based steady-state model of the complete MDI production process is available in Covestro's in-house simulator. The model comprises about 160,000 equations and is an accurate representation of the process that is used for design studies. However, it cannot be directly coupled with an optimization algorithm and its execution time is too long for real-time applications. As proposed in [4], artificial neural networks (ANNs) can be used as a surrogate model of the flowsheet simulation. For the training of the ANN model, the simulator acts as a data generator.

In this case, several ANNs are combined to represent the full plant that is presented in Figure 7a. All quantities of interest, which are the heat duties or steam pressures in the heat exchangers, are modelled individually for each subsystem as represented formally by Equations (43) and (44). The quantities, X , in the reaction stage depend only on the temperature set-point $T_{I,i}$ and the load $\dot{m}_{I,i}$ of the corresponding i -th heat exchanger. For the separation stage, the independent variables are the corresponding temperature set-points $T_{II,i}$, the mixing temperature T_M , and the load $\dot{m}_{II,i}$. T_M is the temperature of the mixed outlets streams that enter the separation stage. The load describes the mass flow entering the unit operation. In the remainder of the work, the load argument is dropped for better readability.

$$X_{I,i} = f(T_{I,i}, \dot{m}_{I,i}) \quad (43)$$

$$X_{II,i} = f(T_{II,i}, T_M, \dot{m}_{II,i}) \quad (44)$$

$$X_{j,i} \in \{p_{VP,j,i}, \dot{Q}_{j,i}\} \quad (45)$$

The manipulated variables (e.g., temperature set-points) and the plant load were sampled on an equidistant grid and the outputs X calculated by the flow-sheet simulator. All models were identified using the Levenberg–Marquardt algorithm in Matlab’s deep learning toolbox [36]. The networks consist of one layer with 8–32 neurons where the *tanh*-activation function and a linear output layer are employed. Further information on the training of the surrogate models can be found in [4].

3.2. Operator Training Simulator

As a representation (or digital twin) of the real plant, an operator training simulator (OTS) is used. Operator training simulators are utilized to train new plant operators and to enhance the abilities of plant operators to handle abnormal operating conditions, which leads to less activation of interlocks and less unplanned shutdowns. The OTS is also utilized to test new process control schemes. This has the advantage that the tests can be performed independent of the production schedule and can be carried out under specified and repeatable conditions. Furthermore, the obtained simulation results can be used to increase the understanding and acceptance of advanced process control methods among plant personnel.

The OTS consists of an emulated version of the distributed control system of the real plant and a semi-rigorous dynamic process model. The dynamic process model is defined and implemented in a manner such that the OTS is able to simulate the process in a numerically stable way for a large set of operating conditions including start-up and shutdown scenarios in real time. The accuracy must be sufficient to make the operators feel as if they control the real plant. The OTS of the MDI plant is realized in the Workforce Competency framework by Honeywell Forge. On the one hand, the OTS model contains much more detail of the plant (e.g., controllers) and on the other hand, the model of the physics and chemistry of the plant is simpler than the stationary model in the flowsheet simulator; there is a significant mismatch between the stationary behaviour of the OTS and the flowsheet simulator. In addition, noise is present in the simulation. This is shown in detail in [4].

4. Application to the Process

The MAWQA scheme presented in Section 2.3 is applied to the MDI production process (Section 3). The OTS is used as a substitute of the real plant as experiments are possible with the OTS and the operators accept it as a faithful representation of the behaviour of the plant. The computed set-points were entered manually to the OTS after every iteration of the MAWQA scheme and the average time to steady-state is 2–4 h. For the initialization of the

algorithm, a case-specific probing sequence is used, which is described in Section 4. Two different scenarios are investigated. In the first scenario, the RTO algorithm is started from a standard operating point of the plant to find optimal operation conditions that minimize the fouling in the heat exchangers. After the algorithm has converged, the total load of the plant is changed, which acts as a known disturbance. In the second trial, the algorithm is started from a different operating point and the process is disturbed after convergence by a change in the heat transfer coefficients of a subset of the heat exchangers, which represents an unmeasured change in the behaviour of the plant. In between the two experiments, the OTS was updated by Covestro in the course of regular maintenance, which leads to small differences in the initial steps of the algorithms. In all experiments, the same formulation of the optimization problem ((46)–(51)) is used and the required models for $\dot{Q}_{j,i}$ and $p_{VP,j,i}$ are identified as described in Section 3.1.

$$\min_{\mathbf{T}_I, \mathbf{T}_{II}, T_M} \sum_{i=1}^{N_I} (p_{VP,I,i} - p_{VP,I,i}^{clean})^2 + \sum_{i=1}^{N_{II}} (p_{VP,II,i} - p_{VP,II,i}^{clean})^2 \quad (46)$$

$$s.t. \dot{Q}_{j,i} \leq 0, i = 1, \dots, N_j, j = \{I, II\} \quad (47)$$

$$\dot{Q}_{tot}^{lb}(\dot{m}_{tot}) \leq \left(\sum_{i=1}^{N_I} \dot{Q}_{I,i} + \sum_{i=1}^{N_{II}} \dot{Q}_{II,i} \right) \leq \dot{Q}_{tot}^{ub}(\dot{m}_{tot}) \quad (48)$$

$$p_{VP,j,i} \leq p_{VP,j,i}^{max}, i = 1, \dots, N_j, j = \{I, II\} \quad (49)$$

$$\mathbf{T}_j^{lb} \leq \mathbf{T}_j \leq \mathbf{T}_j^{ub}, j = \{I, II\} \quad (50)$$

$$T_M = \mathbf{w}_I^T \mathbf{T}_I \quad (51)$$

$\mathbf{T}_I$ and $\mathbf{T}_{II}$ are the set-points of the heat exchangers of the reactors and of the separators. T_M is the temperature of the stream that enters the separation section, which acts as a coupling variable. T_M is based on a mixing rule and is determined by Equation (51). The vector of weighting factors $\mathbf{w}_I$ depends on the heat capacities, mass flows, and temperatures of the outgoing streams of all reactors. The function $h(\bullet)$, cf. Equation (22), is thus linear. The subscripts I identifies the reactors and the subscript II identifies the separators. The differences of the stream pressures in the heat exchangers $p_{VP,j,i}$ and the desired reference values $p_{VP,j,i}^{clean}$ are penalized quadratically. The numbers of reactors and separators are N_I and N_{II} . $\dot{Q}_{j,i}$ are the heat duties, and the total heat duty of all reactors and separators is bounded by the lower bound $\dot{Q}_{tot}^{lb}$ and upper bound $\dot{Q}_{tot}^{ub}$. The bounds depend on the total load of the production plant $\dot{m}_{tot}$. In addition, there is an upper bound p_{VP}^{max} of the available steam pressure. All temperature set-points must be in the range between $\mathbf{T}_j^{lb}$ and $\mathbf{T}_j^{ub}$. All optimization problems were solved in Matlab [36] using IPOPT [38] within the OPTI toolbox [37].

Initial Probing Sequence

Before bias and gradient correction are applied in MA or MAWQA, the system must be probed to collect enough data points to allow estimation of the plant gradients. Initially, finite differences are used; thereafter, quadratic approximations provide more robust gradient estimates. The structure of the MDI-plant can be exploited to reduce the length of the initial probing sequence, as the probing actions for the reaction and separation stages can be decoupled. Ref. [13] proposed maximizing the inverse condition number $\kappa^{-1}(\bullet)$ to yield a good approximation by finite differences. An optimization problem is solved to compute probing actions when the identification matrix S_k becomes nearly singular; in this work, the initial probing sequence also was determined by solving the optimization problem

$$\max_{\Delta_{T_I}^k, \Delta_{T_{II}}^k} \kappa^{-1}(\mathbf{S}_k) \quad (52)$$

$$\text{s.t. } \Delta_{T_j, \min}^k \leq \Delta_{T_j}^k \leq \Delta_{T_j, \max}^k, k = \{1, 2\}, j = \{I, II\} \quad (53)$$

$$T_M^k = \mathbf{w}_I^T \mathbf{T}_I^k, k = \{1, 2\} \quad (54)$$

$$\mathbf{T}_j^k = \mathbf{T}_j^0 + \Delta_{T_j}^k, k = \{1, 2\}, j = \{I, II\} \quad (55)$$

with

$$\mathbf{S}_k = \left[\begin{pmatrix} T_M^0 - T_M^1 \\ \mathbf{T}_{II}^0 - \mathbf{T}_{II}^1 \end{pmatrix} \begin{pmatrix} T_M^0 - T_M^2 \\ \mathbf{T}_{II}^0 - \mathbf{T}_{II}^2 \end{pmatrix} \right]^T. \quad (56)$$

As $N_{z, \max} = 2$, only two perturbations, designated as $\Delta_{T_j}^k$, are required to estimate all plant gradients. The superscript k indicates the perturbation number. The perturbations are added to the initial inputs $\mathbf{T}_j^0$ for the reaction stage (I) and the separation stage (II). The mixing temperature T_M is calculated according to Equation (54). $\Delta_{T_j, \min}$ and $\Delta_{T_j, \max}$ are user-defined parameters that control the minimum and maximum step-size. An exemplary result for the initial probing sequence can be seen in Figure 8. The initial set-points and the two perturbations are plotted for the models in the reaction stage (lower graph) and separation stage (upper graph). The set-points of the separation stage are placed to ensure a good approximation using FD as there are two inputs. After applying two vectors of set-points to the reaction stage, the quadratic approximations can already be computed as the models only have one input. Due to the parallel structure in the reaction stage, only two trajectories are visible as the individual trajectories of the reactors lie on top of each other.

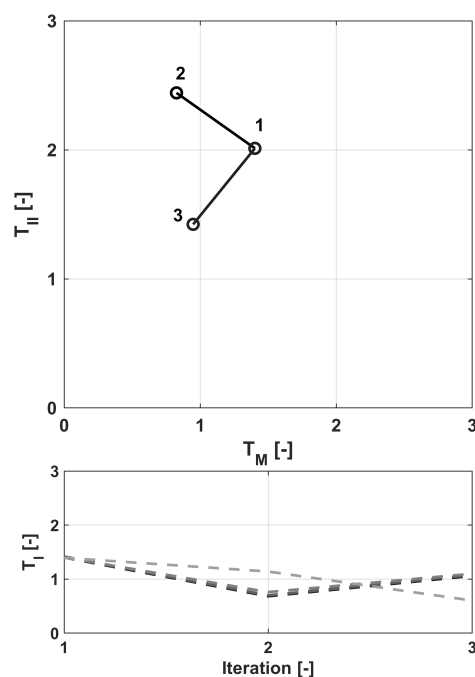


Figure 8. Results of the optimization of the initial probing sequence. In the upper subplot, the scaled temperature set-points of the separation stage T_{II} are plotted on the y-axis and the scaled mixing temperature is plotted on the x-axis. The labels indicate the iteration number. T_M is computed from the set-points of the reaction stage depicted in the lower subplot. The iteration number is displayed on the x-axis and the scaled reaction stage set-points T_I are displayed on the y-axis.

5. Results

Distributed MAWQA, as presented in Section 2.3, was applied in the two different scenarios described above to the OTS of the MDI process as described in Section 3.2.

5.1. Known Disturbance—Load Change

In the load change experiment, the algorithm was started from a known operating point in iteration 1. The resulting trajectories are shown in Figure 9. In iterations 2 and 3, the perturbation sequence according to Section 4 was applied. Due to the reduced number of inputs, quadratic approximations can be constructed for the reaction stage from iteration 4 onward. The gradients for the separation stage were calculated via FD. The set-points of the reaction stage decrease towards 0, which simultaneously decreases the objective function value. The set-points of the separation stage increase for the iterations 5 and 6 and the total heat flow constraint slightly violates the upper bound. In iteration 7, sufficient data points are available to identify quadratic approximations for all subsystems. Due to the improved gradient approximation, the algorithm converges to the optimal operating conditions in iteration 8 satisfying all constraints. In iteration 10, a known disturbance is applied the total load of the plant is reduced by 10%. In Figure 9, this point in time is indicated by the vertical dashed lines. The algorithm is restarted with a new probing sequence similar to the one used before. The new perturbations are applied in the following two iterations. Similar to the first part of the experiment, quadratic approximations are built after iterations 13 to 15 for the reaction section while finite differences are used in the separation section to estimate the plant gradients. From iteration 16 onwards, quadratic approximations are computed for the separation section as well. The algorithm converges to a new optimal set of temperature set-points in iteration 17. Minor oscillations are still visible caused by the noise applied in the OTS.

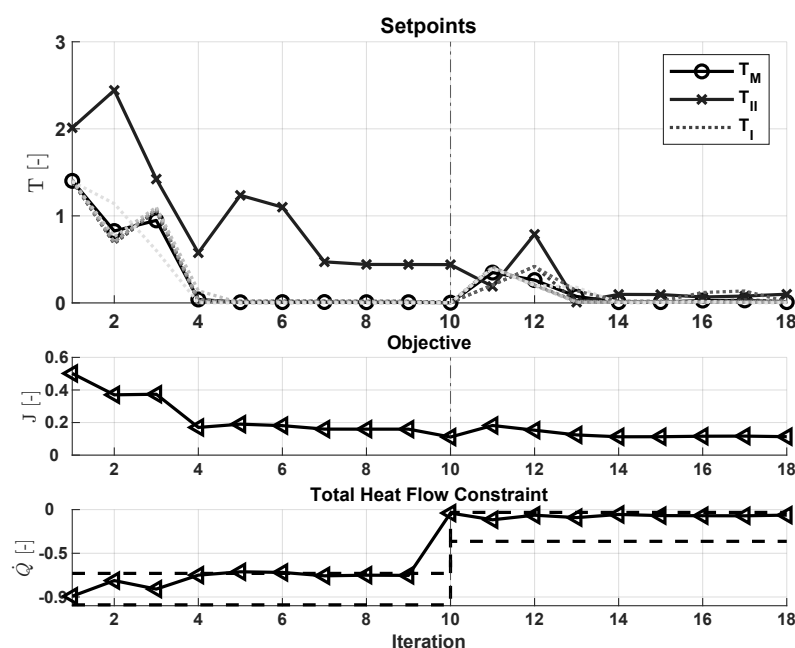


Figure 9. Results of the load change experiment. The scheme is started from a common operating point. After convergence, the total load of the plant was reduced by 10% in iteration 10. In all plots, the iteration number is displayed on the x-axis. In the upper subplot, the temperature set-points of the reactors T_I , the temperature set-points of the separators T_{II} , and the mixing temperature T_M are shown. The second and third subplots show the evolution of the total objective and the global heat duty constraint. All values on the y-axes are scaled. Before the disturbance, initial perturbations are applied during iterations 1–3. Afterwards, finite differences and quadratic approximations are used for each subsystem depending on the number of inputs. The algorithm converges quickly within 8 iterations, once sufficient data points are available to estimate quadratic approximations for all subsystems. When the disturbance occurs, the algorithm restarts and new perturbations are applied to the system. Convergence is again achieved within a similar number of iterations, once quadratic approximations are available for all subsystems.

5.2. Unknown Disturbance—Change in Heat Transfer Coefficients

In this experiment, the algorithm again starts from a standard operating point employing a probing sequence. As in Section 5.1, the algorithm converges quickly once a sufficient number of data points have been collected to identify quadratic approximations for all sub-models. The OTS was updated in between the two experiments, which led to different noise levels and operating conditions. After reaching optimal operating conditions, the unknown disturbance occurs in iteration 10. The resulting trajectory is shown in Figure 10. No predefined perturbations are applied, and quadratic approximations are identified in all iterations for all models. In contrast to the previous experiments, data points older than 6 iterations are removed from the data set for the QAs. The size of the sliding window was chosen based on the minimum number of data points required to identify quadratic approximations for all submodels. Further investigations into the optimal window size may be required for other situations. The algorithm converges to a new optimal set of temperature set-points in 15 iterations. In iterations 11–16, data points both from before and after the occurrence of the disturbance are used to identify the quadratic approximations. Therefore, the algorithm needs longer to converge. Once all the used data points were collected after the disturbance, more heat duty was distributed to the heat exchangers that are not affected by the disturbance, while the heat duty of the heat exchangers that are affected by the disturbance was reduced. This results in a lower value in the objective value and reduces the fouling.

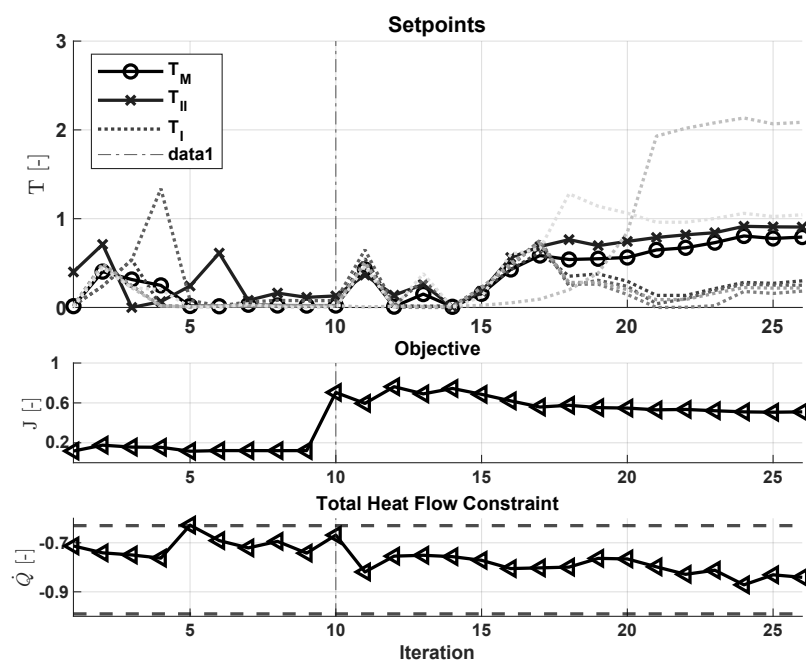


Figure 10. Results of the experiment with an unknown disturbance as described in the text. The scheme is started from a standard operating point. After reaching an optimal steady-state, a disturbance occurs after iteration 10. In all plots, the iteration number is displayed on the x-axis. In the upper subplot, the temperature set-points of the reactors T_I , the temperature set-points of the separators T_{II} , and the mixing temperature T_M are shown. The second and third subplots show the evolution of the total objective and the global heat duty constraint. All values on the y-axes are scaled. The algorithm needs approximately 9 iterations to reach the optimal operating points. Note that higher noise levels are present compared to Section 5.1 due to the updated OTS. In this scenario, only the last six data points are used to approximate the quadratic functions. After the disturbance has occurred in iteration 10, the algorithm converges quickly to the optimal operating point as soon as all data points are collected from the disturbed space. The temperature setpoints are increased in the clean heat exchangers while they are decreased in the fouled heat exchangers.

5.3. Comparison of the Standard MAWQA Algorithm

In [4], the MAWQA algorithm was applied to the MDI process without the distributed estimation of the modifiers. In both cases, artificial neural network models were used in the optimization. The disMAWQA algorithm converges significantly faster to the optimal operation conditions of the plant. Table 2 lists the number of iterations required for convergence. The standard MAWQA algorithm requires twice as many iterations to reach the optimal operating conditions when starting from common operating points. Note that in [4], no external disturbances are applied; therefore, only the iterations until the first convergence are counted for the two scenarios presented in this paper.

Table 2. Comparison of the iterations needed for convergence of the MAWQA and disMAWQA algorithms applied to the MDI process. The standard algorithm [4] needs approximately twice as many iterations compared to the proposed algorithm.

	MAWQA [4]	disMAWQA (Section 5.1)	disMAWQA (Section 5.2)
N_{con}	16	8	9

6. Conclusions and Outlook

In this paper, a modification of the MAWQA algorithm for large-scale plants with few couplings between the subunits is proposed and applied to the MDI production process. By exploiting the structure of the plant in a distributed gradient estimation scheme, the number of required data points that are needed to compute quadratic approximations of the cost and constraint functions, which are the basis of the estimation of the plant gradients, can be drastically reduced. With this modification, the convergence of the MAWQA algorithm is significantly faster compared to previous work, see [4]. The structure of the plant was also exploited to design an initial probing sequence to reduce the needed initial perturbations. The resulting scheme was applied to the operator training simulator (OTS) of the MDI plant in two realistic scenarios. In the first scenario, the load of the plant changed, which represents a known disturbance, while in the second scenario, an unknown disturbance was introduced. In both cases, the algorithm is able to converge to optimal operating conditions fast and smoothly and thereafter to appropriately react to the disturbances.

The distributed estimation makes it possible to also apply MAWQA to larger plants where otherwise, the number of perturbations that are required to estimate the gradients becomes large so that it would be difficult to convince the plant managers to apply RTO with Modifier Adaptation or MAWQA. In the case considered here, the plant structure is parallel-sequential with only one coupling variable. We will investigate which other structures will be promising in the future for MAWQA, with distributed estimation of the modifiers. Also, we assume full knowledge of the relationship of the coupling variables and the manipulated inputs, which is justified in this case, but in general, uncertainties will also be present. How to deal with this issue will be the subject of further work. It may be further beneficial to couple the algorithm with disturbance detection or with a restart algorithm.

7. Patents

The application of iterative RTO to the MDI process has been submitted as a patent [2].

Author Contributions: Conceptualization, J.E., I.W. and S.E.; methodology, J.E. and S.E.; software, J.E.; validation, I.W.; formal analysis, J.E. and S.E.; investigation, J.E.; resources, I.W.; data curation, J.E. and I.W.; writing—original draft preparation, J.E.; writing—review and editing, S.E.; visualization, J.E.; supervision, S.E.; project administration, S.E.; funding acquisition, S.E. All authors have read and agreed to the published version of the manuscript.

Funding: The research that lead to the presented results was part of KEEN project funded by the German Ministry of Economic Affairs and Climate Action (BMWK) under the grant number 01MK20014T.

Data Availability Statement: The datasets presented in this article are not readily available because of confidentiality reasons. Requests to access the datasets should be directed to Covestro Germany AG.

Conflicts of Interest: The authors declare the following potential conflict of interest: All authors are co-inventors of a patent (Patent No. [EP22209645.5]) that is related to the subject matter of this publication. Jens Ehlhardt and Sebastian Engell are employed by TU Dortmund University, and Inga Wolf is employed by Covestro Deutschland AG. Apart from the mentioned patent, the authors declare no additional financial or personal relationships that could have influenced the work reported in this manuscript.

Abbreviations

The following abbreviations are used in this manuscript:

disMAWQA	MAWQA with Distributed Estimation of Modifiers
FD	Finite Differences
KEEN	Künstliche-Intelligenz Inkubator-Labore in der Prozessindustrie
MA	Modifier Adaptation
MAWQA	Modifier Adaptation with Quadratic Approximation
MDA	Methylenedianiline
MDI	Diphenylmethane Diisocyanate
OTS	Operator Training Simulator
QA	Quadratic Approximations
RTO	Real-time Optimization

Appendix A

The least squares problem stated in Equation (A1) is solved to identify the vector of parameters of the quadratic approximation, θ ,

$$\min_{\theta} \sum_{i=1}^{N_{\mathcal{U}}} (J_P(\mathbf{u}_i) - J_{\Phi}(\mathbf{u}_i, \theta))^2, \quad (\text{A1})$$

where $J_P(u_i)$ are the measured values of the objective function and $N_{\mathcal{U}}$ is the number of data points in the regression set $\mathcal{U}$. The regression set is a subset of the available data points and selected to ensure a well poised identification [17]. Analogously, quadratic approximations of the constraint functions G_{Φ} are identified. In addition, more concepts from derivative free optimization are used in MAWQA, for more information see [16]. Depending on the quality of the model and of the quadratic approximation, the surrogate optimization problem ((A2)–(A4)) may be solved instead of ((6)–(8)).

$$\min_{\mathbf{u}^{k+1}} J_{\Phi}(\mathbf{u}^{k+1}) \quad (\text{A2})$$

$$\text{s.t. } \mathbf{G}_{\Phi}^k(\mathbf{u}^{k+1}) \leq \mathbf{0} \quad (\text{A3})$$

$$\mathbf{u}^{lb} \leq \mathbf{u}^{k+1} \leq \mathbf{u}^{ub}, \quad (\text{A4})$$

where J_{Φ} and $\mathbf{G}_{\Phi}$ are the quadratic approximations of the cost and constraint functions. The decision is made based on the following performance metrics, as follows:

$$\rho_{ad}^k = \max \left\{ \left| 1 - \frac{J_{ad}^k - J_{ad}^{k-1}}{J_P^k - J_P^{k-1}} \right|, \left| 1 - \frac{G_{ad,1}^k - G_{ad,1}^{k-1}}{G_{P,1}^k - G_{P,1}^{k-1}} \right|, \dots, \left| 1 - \frac{G_{ad,N_G}^k - G_{ad,N_G}^{k-1}}{G_{P,N_G}^k - G_{P,N_G}^{k-1}} \right| \right\} \quad (A5)$$

$$\rho_{\Phi}^k = \max \left\{ \left| 1 - \frac{J_{\Phi}^k - J_{\Phi}^{k-1}}{J_P^k - J_P^{k-1}} \right|, \left| 1 - \frac{G_{\Phi,1}^k - G_{\Phi,1}^{k-1}}{G_{P,1}^k - G_{P,1}^{k-1}} \right|, \dots, \left| 1 - \frac{G_{\Phi,N_G}^k - G_{\Phi,N_G}^{k-1}}{G_{P,N_G}^k - G_{P,N_G}^{k-1}} \right| \right\}. \quad (A6)$$

In (A5) and (A6), the subscripts ad , P , and Φ are the adapted functions, plant measurements, and quadratic functions, respectively. N_G is the number of constraints. If $\rho_{ad}^k \leq \rho_{\Theta}^k$, problem (6)–(8) will be solved else (A2)–(A4) will be solved. Additionally, a trust-region constraint (A7) is added to (6)–(8) and (A2)–(A4) to ensure the validity of the local quadratic approximations.

$$(\mathbf{u}^{k+1} - \mathbf{u}^k)^T cov(\mathcal{U}^k)(\mathbf{u}^{k+1} - \mathbf{u}^k) \leq \gamma^2 \quad (A7)$$

The trust-region is determined by the regression set $\mathcal{U}$ and a tuning factor γ that controls the size of the trust-region. The performance metrics ρ_{ad} and ρ_{Φ} are used to make the decision between the adapted optimization problem and the optimization based on the quadratic function. In [15], a criticality check is proposed, which is commonly used in derivative free optimization [16]. The criticality step ensures that all points lie close to the next iterate $\mathbf{u}^{k+1}$. In this work, however, the criticality step is not used because oscillating behavior could be observed during tests prior to the experiments.

References

1. Sonnenschein, M.F. *Polyurethanes: Science, Technology, Markets, and Trends*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2021. [CrossRef]
2. Engell, S.; Ahmad, A.; Ehlhardt, J.; Wolf, I.; Arras, J.; Hielscher, A. Method for Controlling a Distributed Control System. EU Patent EP22209645.5, 2022, *applied*.
3. Müller, D.; Dercks, B.; Nabati, E.; Blazek, M.; Eifert, T.; Schallenberg, J.; Piechottka, U.; Dadhe, K. Real-Time Optimization in the Chemical Processing Industry. *Chem. Ing. Tech.* **2017**, *89*, 1464–1470. [CrossRef]
4. Ehlhardt, J.; Ahmad, A.; Wolf, I.; Engell, S. Real-Time Optimization Using Machine Learning Models Applied to the 4,4'-Diphenylmethane Diisocyanate Production Process. *Chem. Ing. Tech.* **2023**, *95*, 1096–1103. [CrossRef]
5. Bárkányi, Á.; Chován, T.; Németh, S.; Abonyi, J. Modelling for digital twins—Potential role of surrogate models. *Processes* **2021**, *9*, 476. [CrossRef]
6. Caballero, J.A.; Grossmann, I.E. An algorithm for the use of surrogate models in modular flowsheet optimization. *AIChE J.* **2008**, *54*, 2633–2650. [CrossRef]
7. Engell, S. Feedback control for optimal process operation. *J. Process Control* **2007**, *17*, 203–219. [CrossRef]
8. Chen, C.Y.; Joseph, B. On-line optimization using a two-phase approach: An application study. *Ind. Eng. Chem. Res.* **1987**, *26*, 1924–1930. [CrossRef]
9. Chachuat, B.; Srinivasan, B.; Bonvin, D. Adaptation strategies for real-time optimization. *Comput. Chem. Eng.* **2009**, *33*, 1557–1567. [CrossRef]
10. Câmara, M.M.; Quelhas, A.D.; Pinto, J.C. Performance evaluation of real industrial RTO systems. *Processes* **2016**, *4*, 44. [CrossRef]
11. Darby, M.L.; Nikolaou, M.; Jones, J.; Nicholson, D. RTO: An overview and assessment of current practice. *J. Process Control* **2011**, *21*, 874–884. [CrossRef]
12. Tatjewski, P. ITERATIVE OPTIMIZING SET-POINT CONTROL—THE BASIC PRINCIPLE REDESIGNED. *IFAC Proc. Vol.* **2002**, *35*, 49–54. [CrossRef]
13. Gao, W.; Engell, S. Iterative set-point optimization of batch chromatography. *Comput. Chem. Eng.* **2005**, *29*, 1401–1409. [CrossRef]
14. Marchetti, A.G.; François, G.; Faulwasser, T.; Bonvin, D. Modifier adaptation for real-time optimization—Methods and applications. *Processes* **2016**, *4*, 55. [CrossRef]
15. Gao, W.; Wenzel, S.; Engell, S. A reliable modifier-adaptation strategy for real-time optimization. *Comput. Chem. Eng.* **2016**, *91*, 318–328. [CrossRef]
16. Conn, A.R.; Scheinberg, K.; Vicente, L.N. *Introduction to Derivative-Free Optimization*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2009. [CrossRef]

17. Wenzel, S.; Yfantis, V.; Gao, W. Comparison of Regression Data Selection Strategies for Quadratic Approximation in RTO. In *Computer Aided Chemical Engineering, Proceedings of the 27th European Symposium on Computer Aided Process Engineering, ESCAPE27, Barcelona, Spain, 1–5 October 2017*; Elsevier: Oxford, UK, 2017; Volume 40, pp. 1711–1716. [\[CrossRef\]](#)
18. Hernández, R.; Dreimann, J.; Engell, S. Reliable Iterative RTO of a Continuously Operated Hydroformylation Process. *IFAC-PapersOnLine* **2018**, *51*, 61–66. [\[CrossRef\]](#)
19. Gottu Mukkula, A.R.; Kern, S.; Salge, M.; Holtkamp, M.; Guhl, S.; Fleicher, C.; Meyer, K.; Remelhe, M.P.; Maiwald, M.; Engell, S. An application of modifier adaptation with quadratic approximation on a pilot scale plant in industrial environment. *IFAC-PapersOnLine* **2020**, *53*, 11773–11779. [\[CrossRef\]](#)
20. Gottu Mukkula, A.R.; Valiauga, P.; Fikar, M.; Paulen, R.; Engell, S. Experimental real time optimization of a continuous membrane separation plant. *IFAC-PapersOnLine* **2020**, *53*, 11786–11793. [\[CrossRef\]](#)
21. Cegla, M.; Fage, A.; Kemmerling, S.; Engell, S. Experimental Application of Real-Time Optimization with Modifier Adaptation and Quadratic Approximation to a Reactive Extrusion Process. *IFAC-PapersOnLine* **2023**, *56*, 6150–6155. [\[CrossRef\]](#)
22. Ahmad, A.; Paulen, R.; Valo, R.; Fikar, M.; Engell, S. Iterative real-time optimization of a membrane pilot plant. *Control Eng. Pract.* **2024**, *147*, 105907. [\[CrossRef\]](#)
23. Schneider, R.; Milosavljevic, P.; Bonvin, D. Distributed modifier-adaptation schemes for the real-time optimisation of uncertain interconnected systems. *Int. J. Control* **2019**, *92*, 1123–1136. [\[CrossRef\]](#)
24. Milosavljevic, P.; Marchetti, A.G.; Cortinovis, A.; Faulwasser, T.; Mercangöz, M.; Bonvin, D. Real-time optimization of load sharing for gas compressors in the presence of uncertainty. *Appl. Energy* **2020**, *272*, 114883. [\[CrossRef\]](#)
25. Papasavvas, A.; Francois, G. Internal Modifier Adaptation for the Optimization of Large-Scale Plants with Inaccurate Models. *Ind. Eng. Chem. Res.* **2019**, *58*, 13568–13582. [\[CrossRef\]](#)
26. Krishnamoorthy, D. A distributed feedback-based online process optimization framework for optimal resource sharing. *J. Process Control* **2021**, *97*, 72–83. [\[CrossRef\]](#)
27. Krishnamoorthy, D.; Skogestad, S. Real-Time optimization as a feedback control problem—A review. *Comput. Chem. Eng.* **2022**, *161*, 107723. [\[CrossRef\]](#)
28. Dirza, R.; Matias, J.; Skogestad, S.; Krishnamoorthy, D. Experimental validation of distributed feedback-based real-time optimization in a gas-lifted oil well rig. *Control Eng. Pract.* **2022**, *126*, 105253. [\[CrossRef\]](#)
29. Ferreira, T.D.A.; Shukla, H.A.; Faulwasser, T.; Jones, C.N.; Bonvin, D. Real-Time optimization of Uncertain Process Systems via Modifier Adaptation and Gaussian Processes. In *Proceedings of the 2018 European Control Conference, ECC 2018, Limassol, Cyprus, 12–15 June 2018*; pp. 465–470. [\[CrossRef\]](#)
30. Andersson, L.E.; Imsland, L. Real-time optimization of wind farms using modifier adaptation and machine learning. *Wind Energy Sci.* **2020**, *5*, 885–896. [\[CrossRef\]](#)
31. Turan, E.M.; Lia, S.; Matias, J.; Jäschke, J. Experimental validation of modifier adaptation and Gaussian processes for real time optimisation. *IFAC-PapersOnLine* **2023**, *56*, 1394–1399. [\[CrossRef\]](#)
32. Costello, S.; François, G.; Bonvin, D. A Directional Modifier-Adaptation Algorithm for Real-Time Optimization. *J. Process Control* **2016**, *39*, 64–76. [\[CrossRef\]](#)
33. Wenzel, S.; Riedl, F.; Engell, S. An efficient hierarchical market-like coordination algorithm for coupled production systems based on quadratic approximation. *Comput. Chem. Eng.* **2020**, *134*, 106704. [\[CrossRef\]](#)
34. Ehlhardt, J.; Wolf, I.; Engell, S. Real-time optimization with machine learning models and distributed modifier adaptation applied to the MDI-process. In *Computer Aided Chemical Engineering, Proceedings of the 34th European Symposium on Computer Aided Process Engineering/15th Symposium on Process System Engineering, Florence, ESCAPE34/PSE24, Italy, 2–6 June 2024*; Elsevier: Oxford, UK, 2024; Volume 53, pp. 1981–1986. [\[CrossRef\]](#)
35. 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.* **2010**, *3*, 1–122. [\[CrossRef\]](#)
36. MATLAB, version 9.12.0 (R2022a); The MathWorks Inc.: Natick, MA, USA, 2022. Available online: <https://www.mathworks.com> (accessed on 9 September 2025).
37. Currie, J.; Wilson, D.I. Opti: Lowering the Barrier Between Open Source Optimizers and the Industrial MATLAB User. In *Proceedings of the Foundations of Computer-Aided Process Operations, Tucson, Arizona, 10–14 January 2027*.
38. Wächter, A.; Biegler, L.T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* **2006**, *106*, 25–57. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.