Real-Time Optimization Using Machine Learning Models Applied to the 4,4′-Diphenylmethane Diisocyanate Production Process

Jens Ehlhardt*, Afaq Ahmad, Inga Wolf, and Sebastian Engell

DOI: 10.1002/cite.202200244

In this work, the optimal time-varying allocation of steam in a large-scale industrial isocyanate production process is addressed. This is a problem that falls into the category of real-time optimization (RTO). The application of RTO in practice faces two problems: First the available rigorous process models may not be suitable for use in real-time connected to the process. Second, there is always a mismatch between the predictions of the model and the behavior of the real plant. We address the first problem by training a neural net model as a surrogate to data generated by a rigorous simulation model so that the model is simple to implement and short execution times result. The second problem is tackled by adapting the optimization problem based on measured data such that convergence to the optimal operating conditions for the real plant is achieved.

Keywords: Isocyanate production, Modifier adaptation with quadratic approximation, Real-time optimization, Surrogate models

Received: December 20, 2022; *revised:* March 07, 2023; *accepted:* March 21, 2023

1 Introduction

On the way towards a sustainable chemical industry, a key interest besides being profitable is the optimal energy- and material-efficient operation of chemical processes and plants. In many processes, heat is supplied through steam, which is produced on-site in combined heat and power plants and the thermal energy is introduced into the process via heat exchangers. The heat exchangers are subject to fouling, which reduces their efficiency and increases the energy demand of the process. The rate of fouling depends on the operating conditions. Therefore, there is strong interest in the optimal operation of these large heat exchangers to reduce fouling while providing the required energy for the conversion and separation processes [1].

Model-based real-time optimization is a powerful tool to determine the most efficient operating conditions, which are then implemented as set-points for low-level controllers [2, 3]. A requirement for real-time optimization (RTO) is the availability of an accurate mathematical model of the investigated plant. For large-scale processes, such detailed first-principles-based models are often available in designated software, e.g., flowsheet simulators. However, the computation times and the lack of connectivity often make them impractical for model-based optimization. A solution to this problem is surrogate modeling. Data is generated by the simulators, and then classical regression or machine learning methods (e.g., artificial neural networks) can be used to identify black-box models that represent the process while being computationally inexpensive [4–6].

The identified surrogate models can be used to compute optimal set-points for the plant. However, the underlying first-principles-based model does not predict the behavior of the real plant accurately and surrogate modeling may introduce an additional plant-model mismatch. Consequently, applying the calculated set-points to the plant may lead to suboptimal or even infeasible operation. In the context of real-time optimization, various approaches have been proposed to cope with this plant-model mismatch, most prominently the two-step approach [7, 8] and modifier adaptation [9, 10]. In the two-step approach, selected model parameters are adapted on the basis of the measured responses of the plant. This works well if the model is structurally (largely) correct and the critical parameters are known. In contrast, in modifier adaptation the optimization problem is modified based on the available data, more

[1]Jens Ehlhardt https://orcid.org/0000-0002-0776-3460 (jens.ehlhardt@tu-dortmund.de),
[1]Afaq Ahmad, [2]Inga Wolf, [1]Sebastian Engell

[1]TU Dortmund, Process Dynamics and Operations Group, Department of Biochemical and Chemical Engineering, Emil-Figge-Straße 70, 44227 Dortmund, Germany.

[2]Covestro AG, Process Technology, Kaiser-Wilhelm-Allee 60, 51373 Leverkusen, Germany.

precisely, the gradients of the response with respect to the inputs are estimated. This has the advantage that effects that are not represented in the model at all can be compensated. as well. Both methods rely on measurement data to achieve convergence to the true optimum.

In this work, real-time optimization with modifier adaptation is applied to the isocyanate production process. A detailed flowsheet simulation is used to generate data that is used to build a surrogate model that is then used in RTO. The iterative set-point optimization algorithm converges to the optimal operating conditions of the plant despite substantial plant-model mismatch. In what follows, the process is described in Sect. 2 while the surrogate model is presented in Sect. 3. Thereafter, the modifier adaption scheme is explained in Sect. 4. Sect. 5 and 6 present the application to the isocyanate process and the results. Finally, conclusion and an outlook on further work are given in Sect. 7.

2 The 4,4′-Diphenylmethane Diisocyanate Production Process

In this work, a process stage of the production of 4,4′-diphenylmethane diisocyanate is considered. In the 1930s, the production of MDI became industrially important after the addition polymerization of difunctional isocyanates was discovered by O. Bayer [11]. Methylene diphenyl diisocyanates (MDI) are a group of important feedstocks in the production of polyurethanes with 4,4′-diphenylmethane diisocyanate being the most widely used precursor [12]. Polyurethanes are mainly used in the production of foams with applications ranging from shoes to refrigerators and the automotive industry [12].

A schematic diagram of the considered process is shown in Fig. 1. In the process stage considered, methylenedianiline (MDA) reacts to MDI in several reactors (stage one). Thereafter, MDI is separated from other gaseous reaction products (stage two). Each stage consists of several parallel and/or sequential subunits. In the reaction stage as well as in the separation step, heat is provided to the process by steam via several heat exchangers.

The aim of this work is to distribute the total amount of heat required in a process stage in an optimal way among the available heat exchangers such that the fouling processes on the utility side of the heat exchangers are reduced and cleaning activities with associated shutdowns have to be performed as infrequently as possible [1]. An analysis of historic data showed that the fouling processes are mainly correlated to the vapor pressure and the temperature of the steam. If the steam temperature in a heat exchanger is high, fouling processes are accelerated on the product side of the heat exchangers. In order to be able to still provide the same amount of heat also in this state of increased fouling, the steam temperature has to be increased, which results in even more accelerated fouling such that cleaning activities are soon unavoidable. Instead of transferring the same amount of heat via a heat exchanger that was subject to increased fouling, it is preferable to distribute a part of the heat flow among the heat exchangers with lesser fouling such that the vapor pressures of all heat exchangers stay close to their clean values. In this manner, the period of time before the next cleaning activities are necessary can be extended [1].

Mathematically, the objective of the case study can be formulated as the minimization of the sum of the squared deviations of the current values of the vapor pressure from the vapor pressure of a clean heat exchanger. This leads to the minimization of the objective function J_m:

$$J_m = \Sigma_{i=1}^{N} \left(p_{VP,i} - p_{VP,i}^{clean} \right)^2 \tag{1}$$

where N is the number of heat exchangers, $p_{VP,j}$ is the vapor pressure of heat exchanger i and $p_{VP,i}^{clean}$ is the vapor pressure for a clean heat exchanger. The manipulated variables that are available for optimization are set-points of the process temperatures T_j on the product sides of the heat exchangers. The process temperatures determine the amount of heat that has to be transferred in the heat exchangers of the process stage. In this work, the number of manipulated variables was reduced to two by applying the same set-point to all heat exchangers in the subunits and by setting other set-points to constant values. The inclusion of more manipulated variables is ongoing work.

3 Data-Based Modeling and Implementation

The process stage is modeled by several artificial neural network models (ANNs) instead of a steady-state model based on first principles. All heat exchangers are modeled by individual ANNs and are combined to represent the full model in the optimization. The required data was generated by a first-principles-based steady state model of the complete MDI production process. The first-principles model comprises about 160 000 equations and is available in Covestro's in-house simulator. The manipulated variables are sampled on an equidistant grid and the outputs such as vapor pressures and heat duties are calculated.

The resulting ANNs consist of one-layer perceptrons with 8 to 32 neurons

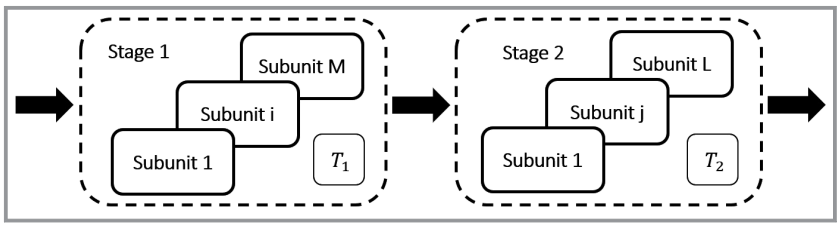

Figure 1. Schematic drawing of the investigated process.

per layer. The *tanh*-activation function is used, and all models are trained using the MATLAB deep learning toolbox [13]. The combined optimization models are depicted in Fig. 2. In Fig. 2a, the objective function is shown. The *x*- and *y*-axes display the two temperature set-points on a normalized scale. The normalized objective value is displayed on the *z*-axis. The model accurately represents the noise-free data points provided by the steady-state simulator. The cost function and the constraint map are depicted in Fig. 2b. The *x*- and *y*-axes again represent the two temperature set-points in a normalized scale. ★ marks the theoretical optimum of the optimization model and the feasible region is indicated by the shaded area. The resulting surrogate model is an accurate representation that can be used for model-based optimization.

4 Real-Time Optimization using Modifier Adaptation with Quadratic Approximation

In the framework of real-time optimization, a steady-state optimization problem (Eqs. (2)–(5)) is solved to calculate the optimal set-points of a process. RTO aims at minimizing an economic goal that is represented by the cost function $J_p$while satisfying n_c constraints G_m. n_u inputs u (e.g., the temperature set-points of the heat exchangers) can be chosen within u^{lb} and u^{ub}, their lower and upper bounds [14].

$$\min_{u} \; J_m(y_m,u) \tag{2}$$

$$s.t. \quad y_m = F_m(u) \tag{3}$$

$$G_m(u) \le 0 \tag{4}$$

$$u^{lb} \le u \le u^{ub} \tag{5}$$

Generally, the steady state plant models $F_m(u)$ and the constraint functions G_m will differ from the real behavior of the plant. This plant-model mismatch may lead to a solution that significantly differs from the true plant optimum or violates the constraints. In RTO, various methods have been developed to cope with the plant-model mismatch and ensure convergence to the true optimum of the process. In the two-step approach, the parameters of the plant model are re-estimated using available measurement data and the updated model is then used to calculate optimal inputs [7, 8]. The two-step approach assumes that the model structure is correct. Furthermore, the number of parameters that can be estimated is limited [9, 15, 16]. Hence, the two-step approach is not applicable when ANNs are used as the optimization model due to their large number of parameters without a physical meaning.

In modifier adaptation [9, 10], the optimization problem itself is iteratively updated by correction terms (the so-called modifiers) based on the measurements. The adaptation of the optimization problem leads to convergence to the true plant optimum even if the structure and the parameters of the model are incorrect [9, 10]. In detail, the adapted optimization problem (Eqs. (6)–(12))

$$\min_{u^{(k+1)}} \; J_{ad}^{(k)}\left(u^{(k+1)}\right) := J_m\left(\hat{y}_m,\; u^{(k+1)}\right) + \lambda_J^{(k)}\left(u^{(k+1)} - u^{(k)}\right) \tag{6}$$

$$s.t. \quad \hat{y}_m = F_m\left(u^{(k+1)}\right) \tag{7}$$

$$G_{ad}^{(k)}\left(u^{(k+1)}\right) := G_m\left(u^{(k+1)}\right) + \varepsilon_c^{(k)} + \lambda_c^{(k)}\left(u^{(k+1)} - u^{(k)}\right) \le 0 \tag{8}$$

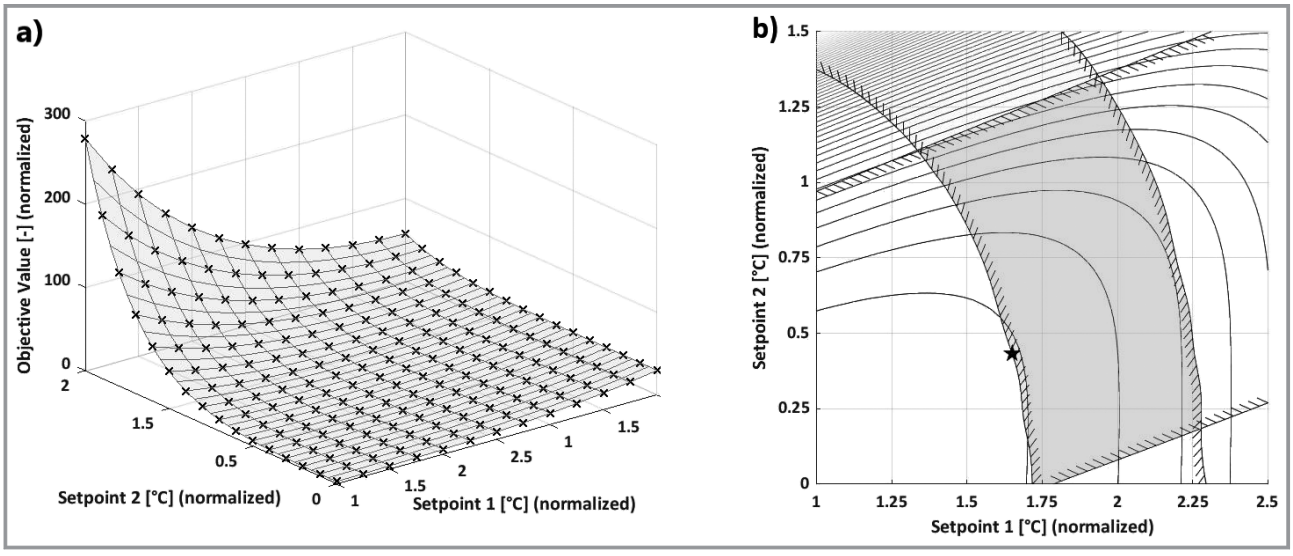

Figure 2. Objective function and constraints with contour plots of the identified models. In both subplots, the temperature set-points are displayed on the *x*- and *y*-axes in a normalized scale. a) The normalized objective value is displayed on the *z*-axis. The model prediction is shown by the mesh while the model data is represented by ×. b) Objective and constraint map of the identified model. ★ represents the theoretical optimum of the identified model and the feasible region is shown as the grey shaded area.

$$u^{lb} \le u^{(k+1)} \le u^{ub} \tag{9}$$

with the correction terms or modifiers

$$\varepsilon_c^{(k)} = G_p^{(k)} - G_m^{(k)} \tag{10}$$

$$\lambda_J^{(k)} = \nabla J_p^{(k)} - \nabla J_m^{(k)} \tag{11}$$

$$\lambda_C^{(k)} = \nabla G_p^{(k)} - \nabla G_m^{(k)} \tag{12}$$

is solved iteratively. Thus, in iteration k the objective and constraints are updated by the zero- and first order modifiers $\varepsilon_c^{(k)}$, $\lambda_J^{(k)}$ and $\lambda_C^{(k)}$. $\nabla J_p^{(k)}$ denotes the gradient of the plant objective function with respect to the inputs u in iteration k and $\nabla G_p^{(k)}$ is the gradient of the plant constraint functions with respect to the inputs in iteration k. Similarly, $\nabla J_m^{(k)}$ and $\nabla G_m^{(k)}$ denote the gradients of the cost and constraint functions of the optimization model. The computed next input $u^{(k+1)}$ is applied to the plant and the optimization is repeated as soon as a new steady state has been reached.

In the optimization problem (6)–(12), the gradients of the objective function of the real plant and of the constraints with respect to the optimization variables are needed. They have to be estimated from the measured data, which is the main challenge in the application of modifier adaptation to real problems. Often, finite difference approximations (FDA) are used [9]. However, FDA are vulnerable to noisy measurement data and finding a suitable step-size that leads to accurate gradients without amplifying the noise or loosing accuracy is challenging. In [14], Gao et al. proposed modifier adaptation with quadratic approximation (MAWQA). MAWQA combines concepts of derivative free optimization and modifier adaptation. It has already been demonstrated successfully for lab- and pilot-scale plants [17–20].

In MAWQA, quadratic approximations (QA) of the objective function and the constraint functions of the real plant are used to calculate the plant gradients $\nabla J_p^{(k)}$ and $\nabla G_p^{(k)}$. To fit quadratic models, a minimum number of $0.5(n_u + 1)(n_u + 2)$ data points are required where n_u is the number of degrees of freedom (inputs that are optimized). After $n_u + 1$ initial perturbations to the process, the modifiers are determined based on finite differences until enough data points have been collected. The parameters θ of the quadratic approximation are then determined via the solution of the least-squares problem (13)

$$\min_{\theta} \Sigma_l^{n_r} \left(J_p(u_l) - J_\Phi(u_l, \theta) \right)^2 \tag{13}$$

where $J_p(u_i)$ are the measured values of the objective function, $J_\Phi(u_i, \theta)$ is the quadratic approximation, and n_r is the number of available data points for the approximation. The quadratic approximations of the constraint functions are determined in the same manner. The gradients of the cost function and of the constraints can then be computed analytically from the quadratic models.

The MAWQA algorithm presented in [14] includes also other elements of derivative-free optimization [21]. Further, the optimization problem may be changed to solving (14)–(16) where the fitted quadratic functions are used instead of the adapted cost and constraint functions.

$$\min_{u^{(k+1)}} J_\Phi^{(k)}\left(u^{(k+1)}\right) \tag{14}$$

$$s.t. \quad G_\Phi^{(k)}\left(u^{(k+1)}\right) \le 0 \tag{15}$$

$$u^{lb} \le u^{(k+1)} \le u^{ub} \tag{16}$$

where J_Φ and G_Φ are the quadratic approximations of the objective and constraints. In both cases, as the quadratic approximations are only valid within the vicinity of the last input, a trust region is defined and added as a constraint to the optimization problems (6)–(12) and (14)–(16):

$$\left(u^{(k+1)} - u^{(k)}\right)^T \mathrm{cov}\left(U^{(k)}\right)^{-1} \left(u^{(k+1)} - u^{(k)}\right) \le \gamma^2 \tag{17}$$

γ is a tuning factor for scaling the size of the trust region and needs to be chosen in advance. The decision between problems (6)–(12) and (14)–(16) is based on a quality check in which the prediction errors of $J_{ad}^{(k)}$ and $G_{ad}^{(k)}$ are compared to that of the quadratic functions J_Φ^k and G_Φ^k.

A simplified flow chart of the MAWQA algorithm is shown in Fig. 3. In every iteration, first convergence to a steady is checked. On convergence, the number of available data points is investigated. If enough data points are available, the quadratic approximations are built. Based on the quality check, either the adapted problem (6)–(12) or problem (14)–(16) is solved to calculate the next inputs. If not enough data points are available for the QA but for FDA, the modifiers are determined using finite difference approximations and the adapted problem is solved. If less than $(n_u + 1)$ data points are available, the plant set-points are perturbed to generate new measurements. The scheme is repeated until the plant has converged to the optimal set-points. The interested reader is referred to [9] and [14] for more information and mathematical background of modifier adaptation using finite difference approximations and MAWQA.

5 Application to the MDI-process

As the MDI production process is usually operated at a steady state over longer periods of time, the process is suited to apply the iterative RTO scheme presented in Sect. 3. In what follows, an operator training simulator (OTS) serves as a representation of the real plant, i.e., the OTS is used as a digital twin to predict measurements from the real plant. Operator training simulators are usually applied to train

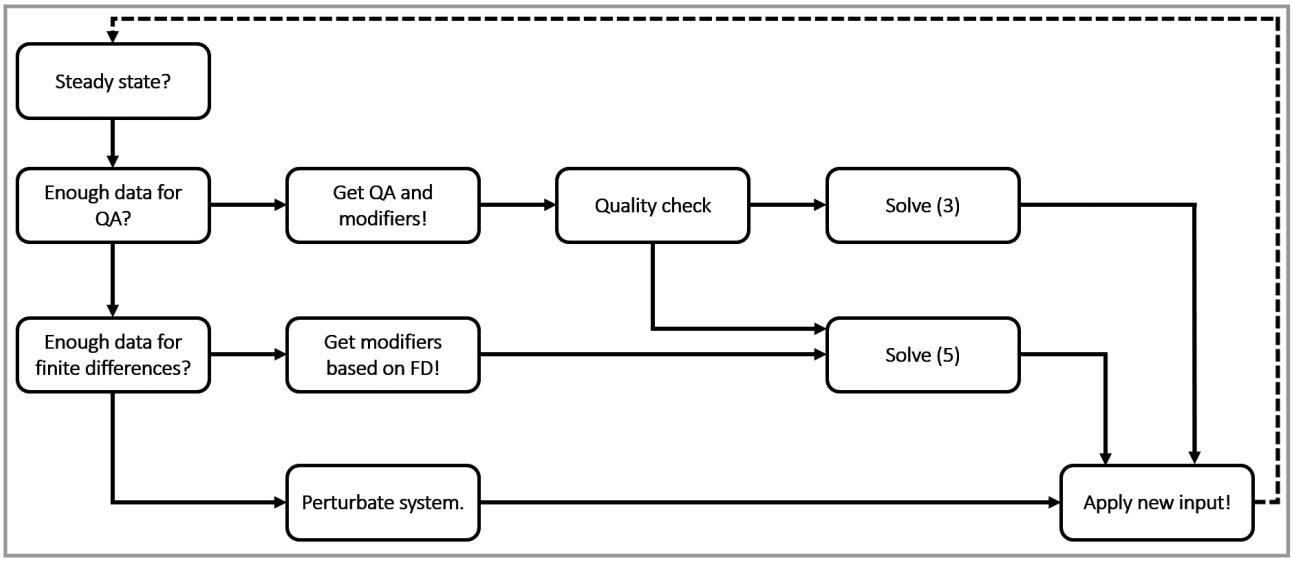

Figure 3. Schematic flow chart of real-time optimization using modifier adaptation with quadratic approximation.

new plant operators and to improve the awareness of all plant operators of abnormal operating conditions that can lead to seldomly encountered failures and triggering of interlocks. Furthermore, an OTS can be run independently from the production schedule and simulations can be carried out under specified and repeatable conditions. Also, the obtained simulation results can be used to increase the acceptance of advanced process control methods among plant personnel. The OTS of the plant considered here was realized in the Workforce Competency framework by Honeywell Forge.

The OTS consists of an emulated version of the distributed control system of the real plant and a rigorous dynamic process model. The rigorous dynamic process model is able to simulate the process in a numerically stable way for a number of predefined operating conditions including start-up and shutdown scenarios in real-time. To achieve this goal, model accuracy might be sacrificed as long as the qualitative behavior of the dynamic process model is represented satisfactorily. For example, simplified reaction kinetics might be used. Therefore, there is a significant model mismatch to the in-house simulator used for the model generation.

The MAWQA scheme is applied to the outputs provided by the OTS and the model developed in Sect. 3 is used as the nominal model in the optimization problem (18)–(22).

$$\min_{T_1,T_2} \quad J_m = \Sigma_{i=1}^{N}\left(p_{VP,i} - p_{VP,i}^{clean}\right)^2 \tag{18}$$

$$s.t. \quad \dot{Q}_i \le 0 \tag{19}$$

$$p_{VP,i} \le p_{VP,i}^{max} \tag{20}$$

$$\dot{Q}_{tot}^{lb} \le \Sigma_{i=1}^{N}\dot{Q}_i \le \dot{Q}_{tot}^{ub} \tag{21}$$

$$T_j^{lb} \le T_j \le T_j^{ub}, \quad j \in \{1,2\} \tag{22}$$

The first constraint (19) assures that the heat exchangers are only considered for heating and not for cooling during the optimization and the second constraint (20) represents an upper bound, $p_{VP,i}^{\max}$, on the vapor pressures that depends on the state of the steam network of the site. The total heat applied to the process stage is bounded to between the values $\dot{Q}_{tot}^{lb}$ and $\dot{Q}_{tot}^{ub}$ by (21). The last constraint (22) imposes lower and upper bounds, T_j^{lb} and T_j^{ub} on the manipulated variables. The nonlinear optimization problems are solved using IPOPT [22] within the OPTI toolbox [23]. In the experiments, the new set-points were given to the OTS which simulated the resulting trajectories. It was identified by inspection that the process had reached a new steady state, which was then fed back to the MAWQA optimization algorithm.

6 Results

The modifier adaptation algorithm presented in Sect. 3 was applied to the simulation of the plant by the OTS. The nominal model in the MAWQA algorithm is the ANN surrogate model that was obtained from fit to the results of the rigorous simulation. Figs. 4 and 5 show the intermediate steps and the final part of the set-point trajectory. The contour plots and the constraints are those that result from the approximation that is used in the specific iteration.

The algorithm starts from a standard operating point (★) and first two initial user-defined perturbations (◆) are implemented in the OTS. The next three set-points are calculated by modifier adaptation based on finite difference approximations. The resulting set-points (marked by ○) stagnate around the theoretical optimum of the optimization model. After six plant evaluations are available, quadratic approximations are built from iteration 7 onwards.

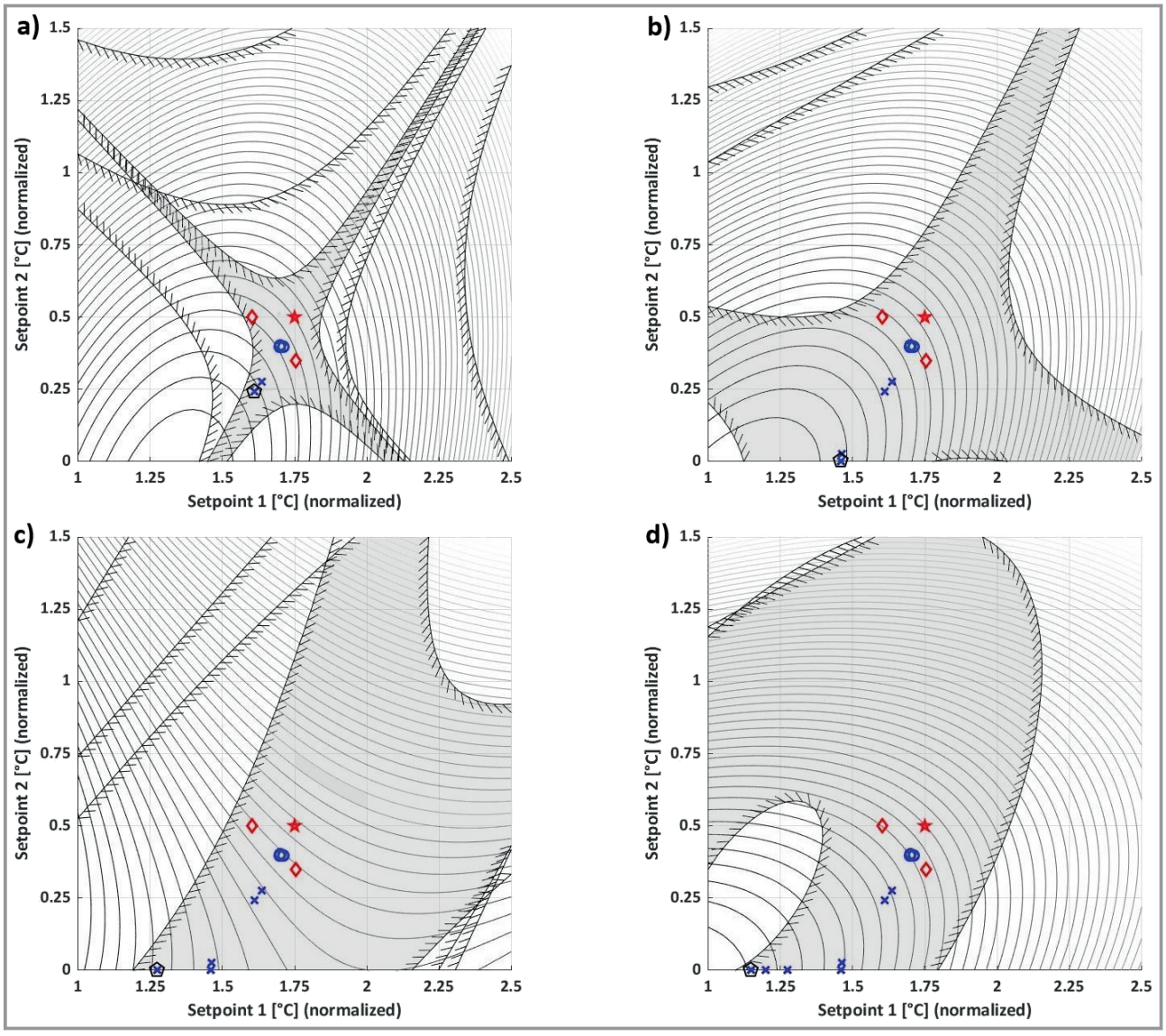

Figure 4. Intermediate iterations of the test run at the OTS. The different subplots a) to d) represent the iterations 9, 11, 13, 15 and the trajectories up to these time steps. In all subplots, the *x*- and *y*-axes indicate the normalized temperature set-points. The initial operating point is depicted by ★. ◆ indicates the initial perturbations. ○ and × represent the set-points that were calculated based on FDA and QA respectively. The results of the fast iterations are indicated by ⌂. The feasible regions are displayed as shaded areas.

In all subsequent iterations, the quality check indicated that the QAs are more accurate than the adapted model, hence problem (14)–(16) based on the quadratic cost function is solved to calculate the inputs from iterations 7 to 20. As soon as the QAs are available, the set-points start to rapidly converge to an optimal setpoint $T \approx [1.2,0]$. It can be noted that the final set-point lies outside the feasible region of the original model but within the feasible region of the adapted model. Fig. 4 shows the objective and constraint maps based on the quadratic approximations of selected intermediate iterations.

It can be seen how the objective and constraint functions evolve during the iterations. The objective function is similar in all iterations and minimum values are expected at minimum values of T_1 and T_2. The constraints can be seen to change drastically in each iteration until they converge to the final approximation shown in Fig. 5. This shows that there is a large mismatch between the constraint functions of the rigorous model and of the OTS. The trust-region constraint is active but not displayed in the plots.

Fig. 6 displays the evolution of the values of the objective function. As long as not enough data points are available for the quadratic approximations, the objective value stagnates around 95. When QAs are available, the value starts to converge to a new optimal value of approximately 60. In iteration 15, the value of the cost function is even lower than the final value. However, not all constraints are fulfilled in the OTS in this iteration. On convergence, all

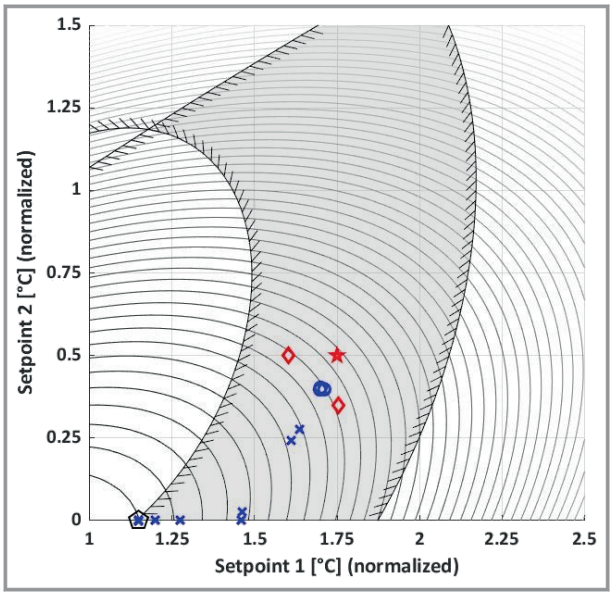

Figure 5. Test run at the OTS. The *x*- and *y*-axis display the normalized temperature set-points. The initial operating point is depicted by ★. The initial perturbations are shown by ◆. ○ and × represent the set-points calculated based on the FDA and QA respectively and the ⌂ marks the optimal set-point. The objective and constraint lines are based on the QA in the last iteration. The feasible region is highlighted by the shaded area.

constraints are fulfilled. It is noticeable that the final objective value is approximately 25 % lower than the optimal value of the surrogate model that is depicted by the dashed line in Fig. 6. On average, the identification of the QA, the determination of the modifiers and the solution of the optimization problems were achieved in less than one second computation time.

7 Conclusion and Outlook

In this work, iterative real-time optimization was successfully applied to the OTS of the 4,4′-diphenylmethane diisocyanate production process to optimize the stream distribution between several heat exchangers. The optimization model is a surrogate model that was identified based on data generated by an in-house flowsheet simulator. Modifier adaption with quadratic approximation was applied to the operator training simulator as a digital twin of the plant using the surrogate model as the nominal model. Despite substantial plant-model mismatch, the iterative optimization converges to the optimal operating conditions. Applying new optimal set-points, the temperature and hence the heat flow in the heat exchangers is reduced. Accordingly, the fouling can be reduced, and the cleaning intervals may be increased [1].

In this work, two temperature set-points have been used in the optimization, multiple parallel heat exchangers were aggregated to sub-units and the overall set-points are optimized. In further work, more individual temperature set-points will be optimized. Furthermore, total load changes and load sharing will be investigated.

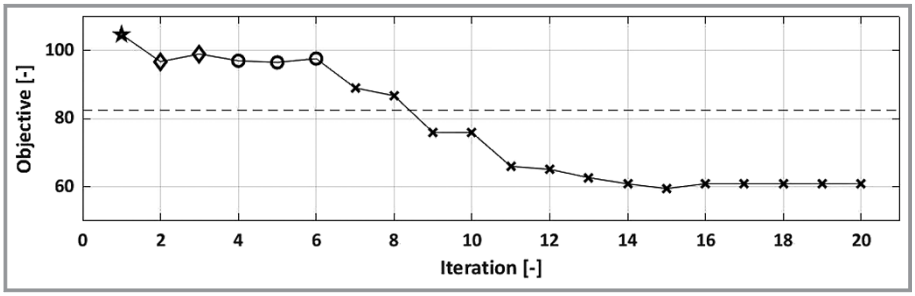

Figure 6. Trajectory of the value of the objective function over the iterations of the RTO algorithm. The initial operating point is depicted by ★. ◆ corresponds to the initial perturbations. ○ and × represent the set-points calculated based on finite difference approximations and quadratic approximations respectively. The theoretical optimum of the optimization model is highlighted by the dashed line.

Acknowledgment

This research has been supported by the project "KI-Inkubator-Labore in der Prozessindustrie – KEEN", funded by the Bundesministerium für Wirtschaft und Klimaschutz (BMWK) under grant number 1MK20014T. This support is gratefully acknowledged. Open access funding enabled and organized by Projekt DEAL.

Symbols used

F	[–]	steady-state function
G	[–]	constraint function
J	[–]	cost function
p	[bar]	pressure
$\dot{Q}$	[kW]	heat-flow
T	[°C]	temperature set-points
u	[–]	input variable
y	[–]	measured variable
U	[–]	regression set

Greek letters

γ	[–]	tuning factor of the trust-region constraint
ε	[–]	zero-order modifier
λ	[–]	first-order modifier
θ	[–]	parameters of the quadratic function
Φ	[–]	quadratic function

Sub- and Superscripts

ad	adapted model
clean	clean heat exchanger
i	heat exchanger index
j	temperature set-point index
lb	lower bound
m	model
max	maximum
k	iteration
ub	upper bound
p	plant
tot	total
VP	vapor pressure
Φ	quadratic function

Abbreviations

ANN	artificial neural network
DFO	derivative free optimization
FDA	finite difference approximations
MAWQA	modifier adaptation with quadratic approximation
MDA	methylenedianiline
MDI	4,4′-diphenylmethane diisocyanate
OTS	operator training simulator
QA	quadratic approximation
RTO	real-time optimization

References

[1] S. Engell, A. Ahmad, J. Ehlhardt, I. Wolf, *Patent EP22209645.5*, **2022**.

[2] D. Müller et al., *Chem. Ing. Tech.* **2017**, *89 (11)*, 1464–1470. DOI: https://doi.org/10.1002/cite.201700033

[3] S. Engell, *J. Process Control* **2007**, *17 (3)*, 203–219. DOI: https://doi.org/10.1016/j.jprocont.2006.10.011

[4] D. Schack et al., *Chem. Ing. Tech.* **2021**, *93 (12)*, 2052–2062. DOI: https://doi.org/10.1002/cite.202100087

[5] A. Bhosekar, M. Ierapetritou, *Comput. Chem. Eng.* **2018**, *108*, 250–267. DOI: https://doi.org/10.1016/j.compchemeng.2017.09.017

[6] G. Brand-Rihm, E. Esche, J.-U. Repke, *Comput.-Aided Chem. Eng.* **2021**, *50*, 611–616. DOI: https://doi.org/10.1016/B978-0-323-88506-5.50097-8

[7] C. Y. Chen, B. Joseph, *Ind. Eng. Chem. Res.* **1987**, *26 (9)*, 1924–1930. DOI: https://doi.org/10.1021/ie00069a034

[8] S.-S. Jang, B. Joseph, H. Mukai, *AIChE J.* **1987**, *33 (1)*, 26–35. DOI: https://doi.org/10.1002/aic.690330105

[9] W. Gao, S. Engell, *Comput. Chem. Eng.* **2005**, *29 (6)*, 1401–1409. DOI: https://doi.org/10.1016/j.compchemeng.2005.02.035

[10] A. G. Marchetti, G. François, T. Faulwasser, D. Bonvin, *Processes* **2016**, *4 (4)*, 55. DOI: https://doi.org/10.3390/pr4040055

[11] *Ullmann's Encyclopedia of Industrial Chemistry*, Wiley-VCH, Weinheim **2000**.

[12] M. F. Sonnenschein, *Polyurethanes: Science, Technology, Markets, and Trends*, 2nd ed., Wiley, Hoboken, NJ **2021**.

[13] *MATLAB (2020b)*, The MathWorks Inc, Natick, MA **2020**.

[14] W. Gao, S. Wenzel, S. Engell, *Comput. Chem. Eng.* **2016**, *91*, 318–328. DOI: https://doi.org/10.1016/j.compchemeng.2016.03.019

[15] W. S. Yip, T. E. Marlin, *Comput. Chem. Eng.* **2002**, *26 (10)*, 1345–1362. DOI: https://doi.org/10.1016/S0098-1354(02)00016-9

[16] A. Ahmad, W. Gao, S. Engell, *Comput. Chem. Eng.* **2018**, *122*, 218–227. DOI: https://doi.org/10.1016/j.compchemeng.2018.08.001

[17] R. Hernandez, J. Dreimann, A. Vorholt, A. Behr, S. Engell, *Ind. Eng. Chem. Res.* **2018**, *57 (26)*, 8750–8770. DOI: https://doi.org/10.1021/acs.iecr.8b00615

[18] A. R. Gottu Mukkula, P. Valiauga, M. Fikar, R. Paulen, S. Engell, *IFAC-PapersOnLine* **2020**, *53 (2)*, 11786–11793. DOI: https://doi.org/10.1016/j.ifacol.2020.12.687

[19] A. R. Gottu Mukkula et al., *IFAC-PapersOnline* **2020**, *53 (2)*, 11773–11779. DOI: https://doi.org/10.1016/j.ifacol.2020.12.685

[20] M. Cegla, S. Engell, *in Proc. Of the 32th European Symp. of Comp. Aided Proc. Eng.* (Eds: L. Montastruc, S. Negny), Elsevier, Amsterdam **2022**.

[21] A. R. Conn, K. Scheinberg, L. N. Vicente, *Introduction to Derivative-Free Optimization*, SIAM, Philadelphia **2009**.

[22] A. Wächter, L. T. Biegler, *Math. Program.* **2009**, *106*, 25–57. DOI: https://doi.org/10.1007/s10107-004-0559-y

[23] J. Curie, D. I. Wilson, *Opti: Lowering the Barrier Between Open Source Optmizers and the Industrial MATLAB User*, **2012**.

