Adaptive Learning of Hybrid Models for Nonlinear Model Predictive Control of Distillation Columns

Jannik T. Lüthje * **Jan C. Schulze** * **Adrian Caspari** *
Adel Mhamdi * **Alexander Mitsos** * **Pascal Schäfer** *

* *Aachener Verfahrenstechnik – Process Systems Engineering, RWTH Aachen University, Aachen, Germany, (e-mail: pascal.schaefer@avt.rwth-aachen.de).*

Abstract:
Our previous work has shown that replacing parts of the classical compartmentalization model reduction approach for distillation columns by offline-trained artificial neural networks (ANNs) improves computational performance. In real-life applications, the absence of a high-fidelity model for data generation can, however, prevent the deployment of this approach. Therefore, we propose a method that utilizes solely plant measurement data, starting from a small initial data set and then continuously adapting to newly measured data. We demonstrate the approach in closed-loop simulations and compare to benchmarks using either the high-fidelity model or an offline trained reduced model for control.

Keywords: Nonlinear model reduction, Model predictive and optimization-based control, Adaptive control, Real time optimization and control, Neural networks in process control

1. INTRODUCTION

Nonlinear model predictive control (NMPC) requires accurate dynamic process models resulting in optimization problems that are typically not solvable in real time. This is particularly the case for distillation columns if applying rigorous full-order stagewise models. Thus, several reduction methods for dynamic models have been developed. A short review of these methods for distillation columns is given in Schäfer et al. (2020). These methods are the collocation approach (Cho and Joseph (1983), applied in Cao et al. (2016)), the compartmentalization approach (Benallou et al. (1986), applied in Schäfer et al. (2019a)), and the wave propagation approach (Marquardt (1989), applied in Caspari et al. (2020)). For a broader review of model reduction methods we refer to Marquardt (2002).

The computational benefit introduced by the model reduction can be further extended by introducing machine learning elements (cf. Schäfer et al. (2020)). In particular, we have shown in previous work that compartmentalized models can be substantially enhanced by substituting stationary stage-to-stage calculations by artificial neural networks (Schäfer et al. (2019a)). In a subsequent work, we demonstrate that this method enables real-time capable NMPC (Schäfer et al. (2019b)). Until now, these hybrid mechanistic/data-driven modeling approaches, however, rely on the availability of large data sets before operation. These data sets are mostly generated offline, utilizing the full-order stagewise model. However, the existence of a full-order model is not guaranteed in real-life application. Thus, recent work has focused on the on-line adaption of the process model to account for an inaccurate initial model for operational purposes (e.g., Tsay et al. (2020)).

In this work, we propose a procedure to continuously adapt our published hybrid model (Schäfer et al. (2019a)) to new measured data, thus, removing the requirement of the existence of a full-order model. To this end, we reformulate the hybrid compartmentalized model as a hybrid stage-aggregation model (cf. Linhart and Skogestad (2010)), which allows for a stricter separation into steady-state blocks to be replaced by artificial neural networks (ANNs) and dynamic single stages with measurable states. As the data-driven model parts have to be continuously adapted to new data, we utilize an on-line learning algorithm that is repeatedly applied, as opposed to a batch learning algorithm that is only applied before operation. We utilize this idea in an in silico control case study from literature and compare to a hybrid control scheme based on offline-generated data and an ideal NMPC using the exact full-order stagewise model.

The remainder of this work is structured as follows: first, we describe the proposed hybrid stage-aggregation model and the implemented adaptive learning algorithm. Next, the case study is presented. Afterwards, we investigate the control performance and real-time applicability of the proposed approach. Finally, an outlook on further work is given.

2. METHOD

2.1 Hybrid stage-aggregation model

The full-order model, which we consider as basis for this work, is taken from Rehm and Allgöwer (1996), where the model is formulated for a distillation column that

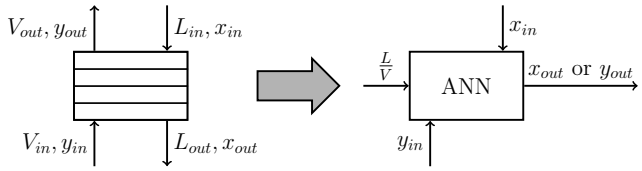

Fig. 1. Replacement of a steady-state column section with an ANN.

is described in Allgöwer and Raisch (1992). It assumes constant molar holdup, ideal thermodynamics, no pressure losses, and no hydrodynamic resistance. These assumptions reduce the model to one mass balance and one thermodynamic equation for each stage (cf. Appendix A).

In a compartmentalization approach (cf. Benallou et al. (1986)), the distillation column is split into multiple compartments. We then assume that the single-stage dynamic behavior can be neglected compared to the overall dynamic behavior of the entire compartment. The final system of equations thus consists of the dynamic compartment balances and steady-state equations for each stage inside the compartment. One stage per compartment does not have a steady-state equation and is referred to as *sensitivity stage*. Thereby, this type of model reduction does not change the steady-state behavior of the column; however, it affects the dynamic response as well as the stiffness of the differential-algebraic system.

For our adaptive approach, we use the stage-aggregation analogy presented by Linhart and Skogestad (2010), reformulating the compartmentalized model in an exact way. That is, the stages of the column are split into two groups, aggregation stages and non-aggregation stages. Aggregation stages have increased holdup, whereas the holdup of non-aggregation stages is set to zero, making them steady-state. The advantage of this approach is that aggregation stages correspond to actual stages of the column, allowing to identify the states of the reduced model from real-world measurements. Formally, this procedure corresponds to multiplying the holdups of aggregation stages with a holdup factor H and reducing the holdup of steady-state trays to zero (cf. Appendix B). Feed stages, the condenser, and the reboiler have to be aggregation stages.

Next, we reduce the total number of equations of the reduced model by replacing sections of steady-state trays, i.e., stationary column sections between two consecutive aggregation stages, with ANNs (Fig. 1), adapting the approach from our previous work considering compartmentalization (Schäfer et al. (2019a)) to stage-aggregation. Note that in contrast to hybrid compartment models, the differential states of hybrid stage-aggregation models are not direct inputs of the surrogate model replacing the stationary parts, leading to generally lower input dimensionalities. Each column section maps the out-going liquid stream of the upper aggregation stage and the out-going vapor stream of the lower aggregation stage to the in-going vapor stream of the upper aggregation stage and the in-going liquid stream of the lower aggregation stage. Due to the simplicity of the assumed full-order model, the liquid and vapor flow, L and V, are constant over the column section. Thus, the ANN does not need to consider the total molar flows as outputs. From the remaining two output concentrations (x_{out} and y_{out}), one can be computed by a mass balance around the entire steady-state column section. Since all trays to be replaced have no holdup, only the ratio of flows, $\frac{L}{V}$, needs to be considered. Thus, the ANNs replacing the column sections have three inputs and one output. Logarithmic scaling is applied to the molar fractions as proposed by Skogestad and Morari (1988).

2.2 Adaptive learning algorithm

There are different interpretations of the definition of online learning in literature (cf. Pérez-Sánchez et al. (2018)). In this work we focus on learning of a continuous data stream. As opposed to batch learning, the complexity of the data is not known beforehand. To ensure an adequate model size at every time step, constructive algorithms are proposed in literature (e.g., Ma and Khorasani (2003)). Constructive algorithms start with a small model size and continuously increase the model size whenever the current model size shows to be incapable of representing the new data.

We implement an adaptive learning algorithm based on the work of Chen and Soo (1993). The overall objective of the algorithm is to improve performance on newly presented data while maintaining performance on previously seen data. For this purpose, all seen data is stored. In each training cycle, a sample of the old data is chosen and mixed with the entire new data set. To select data from the storage, latin hypercube sampling (LHS) is performed in the data region of the data storage. For each sampled point, the nearest neighbor in the data storage is determined. These neighbors finally become part of the training data. With this approach, we build a training data set, which is as equally distributed as possible, out of the unequally distributed data storage. Training of the ANNs is performed with the Levenberg-Marquardt algorithm. The current model is used as a starting point. The balance between improving performance on new data and maintaining performance on previously seen data is tuned by adding a factor to the weights of new data points.

As described earlier, a constructive model building approach is applied to ensure adequate model size at all times. That is, whenever a performance goal cannot be met after the first training cycle, the model size is increased. To this end, the model is restored to its state before the failed training cycle. Then, a new node is added to the single hidden layer. To initialize this new node, a training cycle is performed in which only the weights corresponding to the new node are changed. Considering the linear activation function of the output layer, this is equivalent to fitting a one-node ANN to the prediction error of the old model on the training set. After the new node is initialized, another training cycle is performed using the enlarged model as an initial point, this time adjusting all weights. If the performance goal is still not reached, the procedure is repeated.

2.3 Implementation and case study

The described framework is applied to a distillation column commonly used in literature, first and in most detail

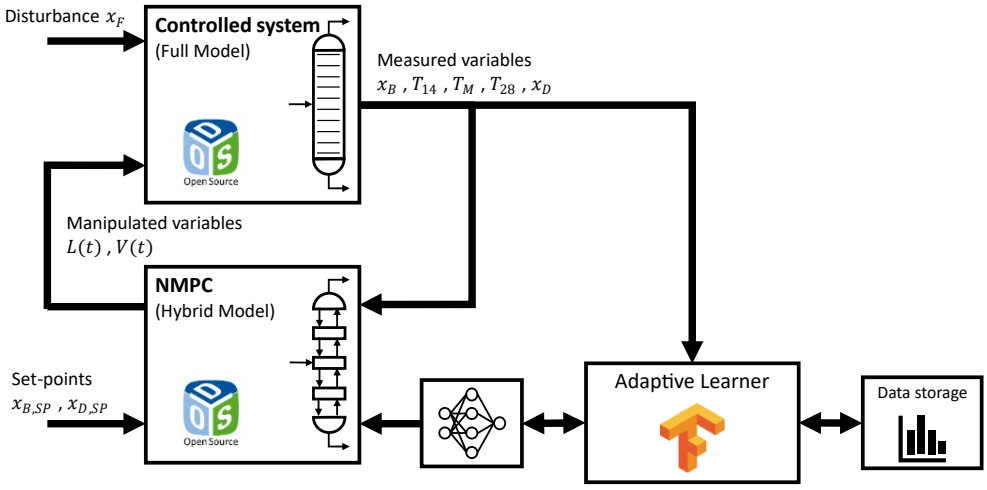

Fig. 2. Implementation of the adaptive control scheme.

described in Allgöwer and Raisch (1992). The column consists of 40 stages and separates a mixture of methanol and propanol. The total molar holdup of the column is assumed to be 10 mol. The control case study is taken from Diehl et al. (2002), meaning that the feed concentration is an unmeasured disturbance. We apply set-points for the molar fractions of both column products. The control objective φ is the integrated sum of the set-point deviations:

$$\varphi = \int_0^{T_\mathrm{P}} \left(x_\mathrm{B,SP} - x_\mathrm{B}\left(t\right)\right)^2 + \left(x_\mathrm{D,SP} - x_\mathrm{D}\left(t\right)\right)^2 \,\mathrm{d}t\,. \quad (1)$$

Table 1. MPC settings

Setting	Symbol	Value
Control horizon	T_C	600 s
Prediction horizon	T_P	1200 s
Control intervals	N	10
Sampling time	T_s	60 s

The MPC settings, summarized in Table 1, are also taken from Diehl et al. (2002), except for the sampling time which is increased from 10s to 60s to allow for a longer solution time of the dynamic optimization problem but still appears appropriate considering the inertia of the column.

Concerning the measurements, Allgöwer and Raisch (1992) specify temperature measurements on stages 14 and 28. As the measurements allow to determine the state of these stages, they are selected as aggregation stages. Additionally, we assume a temperature measurement for the feed stage. Due to the assumption of equilibrium, temperatures in the liquid and gaseous phases of the stages are identical. Finally, we assume concentration measurements for the top and bottom product for the sake of simplicity. Thus, the state of all aggregation stages can be directly derived from the measurements. In a compartmentalization approach this direct identification of the differential variables would not work as compartment states do not correspond to actual column trays. In cases where temperatures are measured only, suitable state estimation techniques can be applied to determine the states of the reduced model.

Training data for the four ANNs can be calculated using mass balances around the aggregation stages. An estimate of the feed flow molar fraction x_F can be determined by a mass balance around the entire hybrid model of the column. Note, that both of these procedures rely on the underlying assumption that the hybrid model matches the full-order model exactly. However, assuming sufficient accuracy of the ANNs, this is only true in the steady-state case. For non-steady-state operation, an error will be made due to lower-order approximation of the dynamics.. To avoid learning wrong data, a weight is added to each data point, which diminishes as the column moves away from steady-state operation.

A closed-loop control framework (cf. Fig. 2) is set up to assess the proposed method. For this, the controlled system is simulated using the full-order stagewise model as plant replacement. The NMPC employs the proposed hybrid stage-aggregation model as controller model. Both the plant replacement and the controller model are implemented in Modelica and made accessible for our in-house software-package DyOS (Caspari et al. (2019)) via export as Functional Mockup Unit (FMU). DyOS enables access to LIMEX (Schlegel et al. (2004)) for state and sensitivity integration and SNOPT (Gill et al. (2005)) for solving the dynamic optimization problems in a single-shooting approach.

The adaptive learner reads the measurement data generated by the simulated column, derives the training data, and adapts the ANNs of the hybrid model. For the sake of simplicity, we assume noise-free measurements. The algorithm is implemented in Python using the Tensorflow library (cf. Abadi et al. (2015)). The interconnections of the described components are also managed in Python. All calculations are performed on an Intel© Xeon© Gold 5117 CPU. The Intel© Optimization for Tensorflow is used.

The adaptive learning algorithm and the NMPC require an initial model to function. For this, we generate open-loop data by performing step tests on the plant replacement. For the step tests, we combine individual step tests on the manipulated variables L and V, and the disturbance x_F. The height of these steps and their time are determined

using latin hypercube sampling. Overall, 20 open-loop trajectories are simulated to build the initial data set. These trajectories are then transformed to training data onto which the ANNs are trained.

The initial ANNs have one hidden layer with four nodes. The tanh activation function is used in the hidden layer while a linear activation function is used in the output layer. All training data is logarithmically transformed as described above and scaled to the range $[-1, 1]$.

Overall, we compare four different approaches for closed-loop control in this work:

(i) The adaptive approach proposed in this work utilizing one data point every 5 s.
(ii) A control scheme based only on the initial model used in the adaptive approach.
(iii) An approach using an offline-trained model from sampled data using the full-order stagewise model as done in previous work. Here, a larger, broader, and more equally distributed data set is used (2000 data points). Also, the data has higher quality, as it is not influenced by the dynamic mismatch between the full-order and the reduced model.
(iv) As a benchmark, we apply the ideal NMPC with full-state feedback and without plant-model mismatch, i.e., the controller model is exact.

3. RESULTS

3.1 Control performance

Fig. 3 presents the results of the four previously described approaches. The non-adaptive approach based only on open-loop data shows strongly oscillating behavior throughout the entire horizon. Large deviations from the set-points can be observed (e.g., the reboiler concentration around $t = 3500\,\mathrm{s}$). The introduction of the adaptive learning algorithm improves the control performance substantially. After $t = 1000\,\mathrm{s}$, almost no oscillating behavior can be observed. Looking at the objective, the integrated sum of squared errors of the set-point deviations, we see a rapid flattening of the trajectory. At the end of the case study, the adaptive approach reaches a control performance comparable to the approach utilizing excessive amounts of offline-generated data. Finally, we remark that the ideal NMPC without plant/model-mismatch allows for only slight improvements over the approach using offline-trained models and thus also over the final performance of the proposed adaptive control scheme.

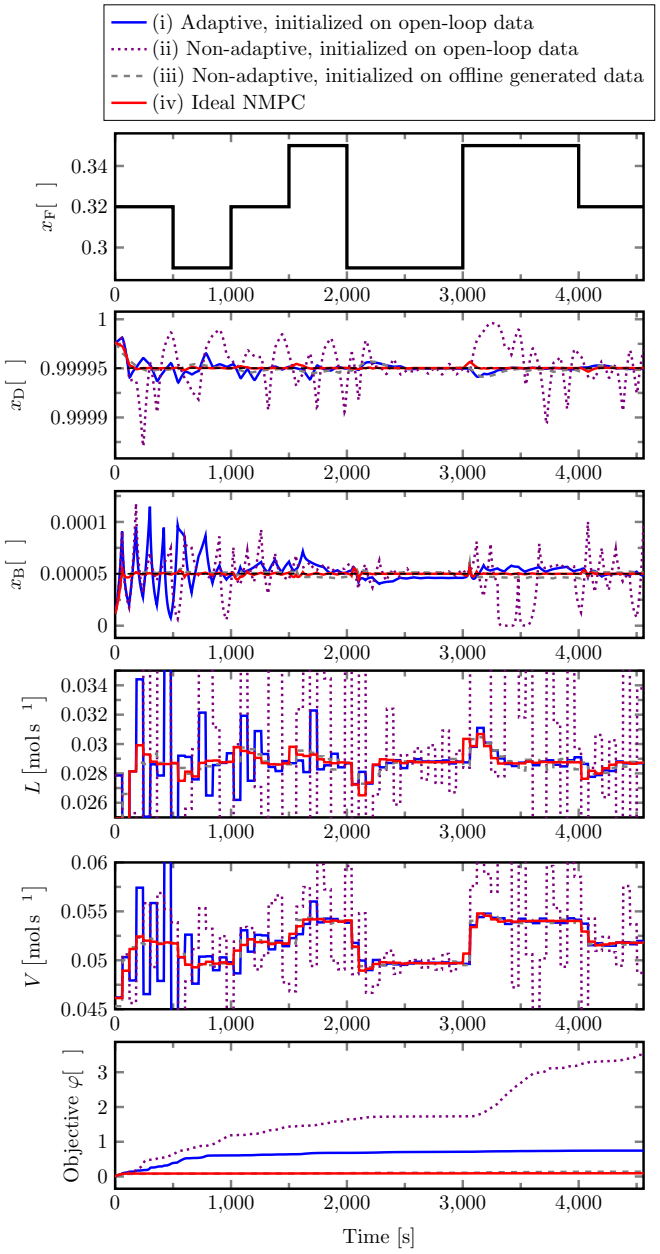

Fig. 3. Comparison of four different control schemes. The upper plot shows the trajectory of the unmeasured disturbance x_F. The remaining plots show trajectories of the plant replacement simulated in closed-loop and trajectories of the manipulated variables. These are the condenser molar fraction x_D, the reboiler molar fraction x_B, the liquid flow rate L, the vapor flow rate V, and the NMPC objective φ. The set-points on x_D and x_B are 0.99995 and 0.00005 respectively.

3.2 Real-time applicability

Beside the control performance, the computational time requirement is another important aspect when rating control schemes, as real-time applicability has to be maintained. For the adaptive approach, the time required to solve the dynamic optimization problem and the time required for adapting the ANNs have to be considered. Note, that the training of the ANNs can be performed in parallel which is not implemented at the moment.

Fig. 4 gives the computational times. The training time shows two large peaks with $t_{Train} > 40\,\mathrm{s}$ at $t = 0\,\mathrm{s}$ and $t = 500\,\mathrm{s}$. At these times large ANN size increases take place. Besides these peaks, the training time stays below 5 s. The dynamic optimization of the adaptive approach finishes in under 15 s throughout the entire case study. The total computational time (sum of training and optimization) only slightly exceeds the sampling time of $T_s = 60\,\mathrm{s}$ once (at $t = 500\,\mathrm{s}$), demonstrating the real-time applicability of the approach in this setting. Concerning the computational time for the dynamic optimization, we

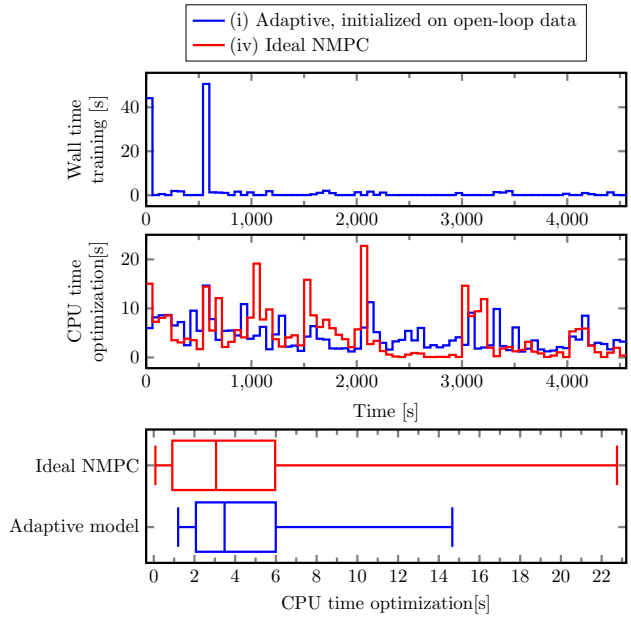

Fig. 4. Overview of computational requirements for the adaptive learning approach and the ideal NMPC. The top plot shows the required time to adapt all ANNs of the adaptive hybrid model. Next, the times required to solve the dynamic optimization problems are presented. The box plot compares the solution times of the dynamic optimizations for the adaptive approach and the ideal NMPC. The minimum, the 25^{th} percentile, the median, the 75^{th} percentile, and the maximum of the data is visualized.

also compare the adaptive approach to the ideal NMPC. To this end, one has to distinguish between two different control scenarios with the first being right after a step in which the disturbance has occurred (cf. x_{F} plot in Fig. 3). Here, a high computational effort is required to solve the dynamic optimization problem as the solution of the previous time step is an inadequate initial guess. For this scenario, we see a large computational benefit of the hybrid model compared to the full-order model used in the ideal NMPC (e.g., at 2000 s), as both approaches require a large number of iterations to solve, thus, allowing the hybrid approach to fully show its computational benefits. In contrast, the second scenario considers times when there is no change in the disturbance. In that case, the lack of plant/model-mismatch of the ideal NMPC and, thus, the existence of better initial guesses leads to a computational advantage for the ideal NMPC during these times, as it requires less iterations to improve the solution. This results in the ideal NMPC having a broader distribution of solution times than the adaptive approach (cf. box plot in Fig. 4). Note however that the high computational times after occurrence of disturbances are the more crucial ones with regard to real-time applicability. Furthermore and following our previous results (cf. Schäfer et al. (2019b)), we suspect computational time savings when applying the hybrid model in the adaptive approach to become even more significant if the control task becomes more complex (e.g., more complex model, more severe disturbances, or economic objectives).

4. CONCLUSION

We develop an adaptive control scheme for distillation columns that continuously trains the data-driven parts of a reduced hybrid controller model based on measurement data. For this purpose, we present a hybridized stage-aggregation approach and implement an adaptive learning algorithm. The approach manages to improve control performance over time, compared to a non-adaptive approach, while maintaining real-time applicability.

In this work, the method is demonstrated in a single-column case study based on a simplified full-order model. Introducing more complexity to the model would result in larger ANNs (e.g., non-constant flows would require at least one additional ANN output). As for the framework, the robustness of the approach could be improved by adding constraints that restrict the dynamic optimization to previously explored data regions. Furthermore, different approaches to deal with the unequally distributed closed-loop data can be investigated. Also, the approach can be applied to the use case of a drift in the underlying process. By implementing a forgetting factor for stored data points, the data set could be replaced over time to match the new process, thus adapting to the changed process behavior.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support of the Kopernikus project SynErgie by the Federal Ministry of Education and Research (BMBF) and the project supervision by the project management organization Projektträger Jülich.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.

Allgöwer, F. and Raisch, J. (1992). *Multivariable controller design for an industrial distillation column.* doi: 10.18419/opus-4404.

Benallou, A., Seborg, D.E., and Mellichamp, D.A. (1986). Dyanmic compartmental models for separation processes. *AIChE Journal*, 32(7), 1067–1078. doi: 10.1002/aic.690320703.

Cao, Y., Swartz, C.L.E., Flores-Cerrillo, J., and Ma, J. (2016). Dynamic modeling and collocation–based model reduction of cryogenic air separation units. *AIChE Journal*, 62(5), 1602–1615. doi:10.1002/aic.15164.

Caspari, A., Bremen, A.M., Faust, J., Jung, F., Kappatou, C.D., Sass, S., Vaupel, Y., Hannemann-Tamás, R., Mhamdi, A., and Mitsos, A. (2019). DyOS - A Framework for Optimization of Large-Scale Differential Algebraic Equation Systems. In A.A. Kiss, E. Zondervan, R. Lakerveld, and L. Özkan (eds.), *Computer Aided*

Chemical Engineering : 29 European Symposium on Computer Aided Process Engineering, volume 46, 619–624. Elsevier. doi:10.1016/B978-0-12-818634-3.50104-1.

Caspari, A., Offermanns, C., Ecker, A.M., Pottmann, M., and Mitsos, A. (2020). A wave propagation approach for reduced dynamic modeling of distillation columns: Optimization and control. *Journal of Process Control*, 91, 12–24. doi:10.1016/j.jprocont.2020.05.004.

Chen, H.W. and Soo, V.W. (1993). Design of adaptive and incremental feed-forward neural networks. In *IEEE International Conference on Neural Networks*, 479–484 vol.1. doi:10.1109/ICNN.1993.298604.

Cho, Y.S. and Joseph, B. (1983). Reduced–order steady–state and dynamic models for separation processes. Part II. Application to nonlinear multicomponent systems. *AIChE Journal*, 29(2), 270–276. doi:10.1002/aic.690290214.

Diehl, M., Bock, H., Schlöder, J.P., Findeisen, R., Nagy, Z., and Allgöwer, F. (2002). Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4), 577–585. doi:10.1016/S0959-1524(01)00023-3.

Gill, P.E., Murray, W., and Saunders, M.A. (2005). SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. *SIAM Review*, 47(1), 99–131.

Linhart, A. and Skogestad, S. (2010). Reduced distillation models via stage aggregation. *Chemical Engineering Science*, 65(11), 3439–3456. doi:10.1016/j.ces.2010.02.032.

Ma, L. and Khorasani, K. (2003). A new strategy for adaptively constructing multilayer feedforward neural networks. *Neurocomputing*, 51, 361–385. doi:10.1016/S0925-2312(02)00597-0.

Marquardt, W. (2002). Nonlinear Model Reduction for Optimization and Control of Transient Chemical Processes. 12–42.

Marquardt, W. (1989). Wellenausbreitung in verfahrenstechnischen Prozessen. *Chemie Ingenieur Technik*, 61(5), 362–377. doi:10.1002/cite.330610504.

Pérez-Sánchez, B., Fontenla-Romero, O., and Guijarro-Berdiñas, B. (2018). A review of adaptive online learning for artificial neural networks. *Artificial Intelligence Review*, 49(2), 281–299. doi:10.1007/s10462-016-9526-2.

Rehm, A. and Allgöwer, F. (1996). Nonlinear H∞-control of a high purity distillation column. In *UKACC International Conference on Control*. IET Digital Library. doi:10.1049/cp:19960720.

Schäfer, P., Caspari, A., Kleinhans, K., Mhamdi, A., and Mitsos, A. (2019a). Reduced dynamic modeling approach for rectification columns based on compartmentalization and artificial neural networks. *AIChE Journal*, 65(5), e16568. doi:10.1002/aic.16568.

Schäfer, P., Caspari, A., Mhamdi, A., and Mitsos, A. (2019b). Economic nonlinear model predictive control using hybrid mechanistic data-driven models for optimal operation in real-time electricity markets: In-silico application to air separation processes. *Journal of Process Control*, 84, 171–181. doi:10.1016/j.jprocont.2019.10.008.

Schäfer, P., Caspari, A., Schweidtmann, A.M., Vaupel, Y., Mhamdi, A., and Mitsos, A. (2020). The Potential of Hybrid Mechanistic/Data-Driven Approaches for Reduced Dynamic Modeling: Application to Distillation Columns. *Chemie Ingenieur Technik*.

Schlegel, M., Marquardt, W., Ehrig, R., and Nowak, U. (2004). Sensitivity analysis of linearly-implicit differential–algebraic systems by one-step extrapolation. *Applied Numerical Mathematics*, 48(1), 83–102. doi:10.1016/j.apnum.2003.07.001.

Skogestad, S. and Morari, M. (1988). Understanding the dynamic behavior of distillation columns. *Industrial & Engineering Chemistry Research*, 27(10), 1848–1862. doi:10.1021/ie00082a018.

Tsay, C., Cao, Y., Wang, Y., Flores-Cerrillo, J., and Baldea, M. (2020). *Identification and online updating of dynamic models for demand response of an industrial air separation unit.*

Appendix A. FULL-ORDER MODEL

Standard tray:

$$n_i \frac{dx_i}{dt} = L^* \left(x_{i,in} - x_i\right) + V \left(y_{i,in} - y_i\right) \tag{A.1}$$

with $L^* = L$ in the rectifying section and $L^* = L + F$ in the stripping section.

Condenser (D):

$$n_D \frac{dx_D}{dt} = V \left(y_{in} - x_D\right) \tag{A.2}$$

Feed tray (M):

$$\begin{aligned} &n_M \frac{dx_M}{dt} \\ &= L \left(x_{in} - x_M\right) + V \left(y_{in} - y_M\right) + F \left(x_F - x_M\right) \end{aligned} \tag{A.3}$$

Reboiler (B):

$$n_B \frac{dx_B}{dt} = \left(L + F\right) \left(x_{in} - x_B\right) + V \left(x_B - y_B\right) \tag{A.4}$$

Thermodynamics:

$$y_i = \frac{\alpha \, x_i}{1 + \left(\alpha - 1\right) x_i} \tag{A.5}$$

Appendix B. REDUCED MODEL

Aggregation stage:

$$H_i n_i \frac{dx_i}{dt} = L^* \left(x_{i,in} - x_i\right) + V \left(y_{i,in} - y_i\right) \tag{B.1}$$

Non-aggregation stage:

$$0 = L^* \left(x_{i,in} - x_i\right) + V \left(y_{i,in} - y_i\right) \tag{B.2}$$

Condenser (D):

$$H_D n_D \frac{dx_D}{dt} = V \left(y_{in} - x_D\right) \tag{B.3}$$

Feed tray (M):

$$\begin{aligned} &H_M n_M \frac{dx_M}{dt} \\ &= L \left(x_{in} - x_M\right) + V \left(y_{in} - y_M\right) + F \left(x_F - x_M\right) \end{aligned} \tag{B.4}$$

Reboiler (B):

$$H_B n_B \frac{dx_B}{dt} = \left(L + F\right) \left(x_{in} - x_B\right) + V \left(x_B - y_B\right) \tag{B.5}$$