Computers and Chemical Engineering 177 (2023) 108272

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Polynomial NARX-based nonlinear model predictive control of modular chemical systems

Anastasia Nikolakopoulou, Richard D. Braatz *

Department of Chemical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA

ARTICLE INFO

Keywords:
Modular systems
Modular chemical systems
Input–output models
Sparse regression
Nonlinear model predictive control

ABSTRACT

The design of control systems for modular chemical systems typically requires the identification of nonlinear dynamic models. Mechanistic models for modular chemical systems are typically of high order, which results in high online computational cost when the models are incorporated into the nonlinear model predictive control (NMPC) formulations developed for explicitly taking constraints into account. This article proposes the use of a particular class of nonlinear input–output models, polynomial nonlinear-autoregressive-with-exogenous-inputs (NARX) models, in the NMPC formulations. A machine learning algorithm, elastic net, is used to select which terms to include within the NARX polynomial series representation. The approach for constructing sparse predictive models and their use in real-time implementable NMPC are demonstrated in a two-input two-output chemical reactor case study. The Julia programming language is used to solve the NMPC optimization problem, resulting in low online computational cost.

1. Introduction

This article considers the model-based control of modular chemical systems, which have become of interest in recent years in both academia and industry, with the potential benefits of reducing supply chain disruptions, production times, costs, waste, and product quality variations, while facilitating faster demand change responses (Mascia et al., 2013; Baxendale et al., 2015; Myerson et al., 2015). Manufacturing in compact modular systems provides additional flexibility in where and when the chemical is produced (Rogers et al., 2020).

Mathematical representations of systems for model-based control can take different forms, which span the spectrum from first-principles and mechanistic models (white-box) to data-driven (black-box) models (von Stosch et al., 2014). Data-driven models include (nonlinear) autoregressive with exogenous inputs ((N)ARX), (nonlinear) autoregressive moving average with exogenous inputs ((N)ARMAX), state–space models, and recursive neural networks (RNN) (Sun and Braatz, 2020). The construction of data-driven models require more data compared to first-principles models, and data-driven models lack extrapolating capabilities and are often uninterpretable. On the other hand, first-principles models require deep knowledge of the process, and their development often takes place over longer time scales (Pearson, 1995; von Stosch et al., 2014). Also, real-time optimal control and model predictive control face computational limitations when based on first-principles models for modular chemical systems (Nikolakopoulou et al., 2019, 2020a).

Data-driven model identification has been widely used in the control of manufacturing systems for decades. Linear input–output (IO) models are typically used for the control of multivariable dynamical systems with actuator, state, and output constraints within model predictive control (MPC) algorithms which have low on-line computational cost. The linearity assumption naturally imposes limitations for processes with strong nonlinearities, especially when the system has a limited number of manipulated variables at the upper level regulatory layer. However, the use of nonlinear models within MPC algorithms has been more limited, due to higher computational costs associated with solving the resulting dynamic optimal control problems.

The main scope of nonlinear model structure selection is to consider model structures that can capture the main characteristics of the process behavior and that do not capture dynamic behavior that the process does not exhibit (Pearson, 2003). The MATLAB System Identification Toolbox (MATLAB System Identification Toolbox, 2019) implements a large number of traditional methods for nonlinear IO model construction. In contrast, Smart Process Analytics (SPA) is software that uses modern methods for automated method selection and model construction for manufacturing data (Sun and Braatz, 2021). The availability of model construction techniques in SPA is informed by expert domain knowledge on machine learning and manufacturing processes, and the method selection is based on the manufacturing data characteristics. The data characteristics that are assessed in SPA

* Corresponding author.
E-mail addresses: anikol@mit.edu (A. Nikolakopoulou), braatz@mit.edu (R.D. Braatz).

https://doi.org/10.1016/j.compchemeng.2023.108272
Received 2 November 2021; Received in revised form 22 March 2023; Accepted 20 April 2023
Available online 27 May 2023

are nonlinearity (between predictors and response), multicollinearity (between predictors), dynamics (of residuals), and heteroscedasticity (of residuals). Process data collected in modular chemical systems are expected to exhibit some degree of nonlinearity and dynamics. For such data sets, SPA recommends the use of recurrent neural networks (RNN) and dynamic algebraic learning via elastic net (DALVEN) for model identification (Sun and Braatz, 2020).

RNNs can model complicated nonlinear dynamical systems with high accuracy and fall into the category of non-interpretable models. Neural-network-based NMPC has been applied in the chemical industry for decades (Qin and Badgwell, 2003). RNNs and in general black-box models require a large quantity of training data to avoid overfitting which can be limiting for manufacturing data sets. Another drawback of neural network models is that their mathematical structure makes it challenging to derive methods for the nonconservative analysis of closed-loop stability and performance (Nikolakopoulou et al., 2020b, 2021).

DALVEN, on the other hand, uses the form of a NARX model (Billings, 2013). DALVEN constructs nonlinear transformations of input variables (past inputs and outputs of the process), combines them in a linear-in-the-parameters model, and uses elastic net (EN) to build a sparse model, which provides interpretability (Sun and Braatz, 2020). Interpretable models can be more reliable when the number of training samples is limited (Sun and Braatz, 2021) which is the case, for example, for continuous pharmaceutical manufacturing data. Several types of nonlinear transformations are considered in DALVEN for each input variable i, such as x_i, $\sqrt{x_i}$, $\ln x_i$, $1/x_i$, as well as up to third-order interactions between them.

Nonlinear dynamical systems can often be written exactly or be approximated as polynomial systems using Taylor series (Harinath et al., 2019) and this commonly occurs in applications such as chemical reactors and biological networks. As such, polynomial NARX-based models are promising model structures for the mathematical description of modular chemical systems. Polynomial models maintain the advantage of higher interpretability compared to neural network models. Additionally, polynomial models have a mathematical structure that is more amenable to the development of control systems analysis and design algorithms. Convex but suboptimal feedback control formulations are restricted to particular classes of polynomial systems, leaving space for exploring MPC solutions (Harinath et al., 2019). Recent advances made in polynomial programming for static systems have been extended to nonlinear dynamical systems to derive nonconservative stability analysis conditions and to formulate NMPC algorithms (Harinath et al., 2019). In general, polynomial optimization is nonconvex and NP-hard, but polynomial solvers can now provide solutions in polynomial time when series of "relaxed" optimizations are solved instead (Henrion and Lasserre, 2002; Prajna et al., 2002). While polynomial models for NMPC are derived in Harinath et al. (2019) using Taylor's theorem, polynomial structures are generated in this work by focusing on nonlinear polynomial transformations in a NARX framework.

A full expansion of a polynomial NARX structure results in a large number of regressors, many of which are not insignificant; which regressors are significant is not known *a priori*. Therefore, building a parsimonious polynomial NARX model for control becomes a problem of model structure selection. Machine learning methods have been applied for this purpose. The least absolute shrinkage and selection operator (lasso) has been used for the sparse identification of NAR(MA)X models (Kukreja et al., 2006; Ribeiro and Aguirre, 2018) and high-dimensional time series estimation (Basu and Michailidis, 2015), and a variation called "square root lasso" has been implemented for the identification of multivariate posynomial models (Calafiore et al., 2015). Elastic net has been used for the identification of nonparametric model structures for linear time-invariant discrete-time systems (Calafiore et al., 2017) and for the construction of sparse industrial process models (Chiu and Yao, 2013; Severson et al., 2015).

The coupling of nonlinear system identification methods with predictive control has been explored in many past studies. A study on receding horizon predictive control based on NARX models employed a conversion to a state–space model to establish closed-loop stability results (De Nicolao et al., 1997). An NMPC algorithm based on polynomial ARX models was formulated in which the future output predictions were rewritten in terms of the control inputs, thus resulting in an objective function which was a polynomial in all of the control inputs (Sriniwas and Arkun, 1997). Polynomial NARMAX models with approximate predictors have been used in NMPC schemes (Bai and Coca, 2007), where the control moves participated in the NARMAX model and no nonlinear terms with respect to the future control moves were assumed. ARX-based MPC in state–space form has been applied to an evaporator model (Huusom and Jørgensen, 2014), by fitting process data to a simple linear structure. NMPC for a distillation column has been implemented using polynomial NARX models, which assumed predefined polynomial model structures (Sriniwas et al., 1995).

In this work, we build on our recently published results on polynomial NARX NMPC for distributed parameter systems (Nikolakopoulou and Braatz, 2022), and demonstrate how polynomial IO NARX predictive models are used in NMPC to control modular chemical systems with multiple inputs and outputs. No prior assumptions on the polynomial NARX models' structure are made. Elastic net (EN) is used to promote variable selection and model sparsification to increase interpretability and reduce the model bias. The NMPC implementation results in a real-time implementable algorithm. The latter task is challenging; NMPC formulations are expected to be nonconvex in the general case making their solution computationally expensive limiting their applicability.

The rest of this article is organized as follows. Section 2 presents some background on polynomial NARX models and machine learning algorithms useful for their identification. Sections 3 and 4 discuss the methodologies for constructing the predictive models and implementing a polynomial NARX-based NMPC respectively. A case study that demonstrates the successful implementation of the methodology is presented in Section 5, followed by the conclusions in Section 6.

2. Background

A single-input single-output (SISO) nonlinear dynamical system can be represented under certain conditions by the difference equation (Billings and Voon, 1986)

$$y(k) = f[u(k-d-M+1), \dots, u(k-d), y(k-M_y), \dots, y(k-1)] + \epsilon(k), \quad (1)$$

where $f[\cdot]$ is a nonlinear function of the system input $u(\cdot)$, $y(\cdot)$ is the system output, d is the time delay, and M and M_y are positive integers that indicate the number of past inputs and outputs included in the model. Eq. (1) describes the class of NARX systems and can be easily extended to the multiple-input multiple-output (MIMO) case. Models commonly used in system identification such as the Hammerstein, Wiener, and Volterra series models are special cases of (1). An equivalent representation for nonlinear stochastic systems can be derived to give the nonlinear autoregressive moving-average model with exogenous inputs (NARMAX) (Billings and Voon, 1986)

$$\begin{aligned} y(k) = g[&u(k-d-M+1), \dots, u(k-d), y(k-M_y), \dots, y(k-1), \\ &\epsilon(k-M_\epsilon), \dots, \epsilon(k-1)] + \epsilon(k), \end{aligned} \quad (2)$$

where $\epsilon(k)$ is the prediction error and $g[\cdot]$ is a nonlinear map. Restricting $f[\cdot]$ or $g[\cdot]$ to multivariable polynomials has the advantage that linear regression techniques can be used, since the model would be linear-in-the-parameters.

In this work, the focus is on models of the structure (1) with $d = 1$, which is not limiting if M is selected to be large enough. The nonlinear function f is a polynomial function of order r, meaning that the arguments of f participate in monomials and polynomials of order

up to and including r in every possible combination. For example, for the SISO model

$$y(k) = f[u(k-3), u(k-2), u(k-1), y(k-2), y(k-1)], \tag{3}$$

a polynomial order $r = 2$, (3) yields

$$\begin{aligned} y(k) =& u(k-3) + u(k-2) + u(k-1) + y(k-2) + y(k-1) + u(k-3)^2 \\ &+ u(k-2)^2 + u(k-1)^2 + u(k-3)u(k-2) + u(k-3)u(k-1) \\ &+ u(k-2)u(k-1) + y(k-2)^2 + y(k-1)^2 + y(k-2)y(k-1) \\ &+ u(k-3)y(k-2) + u(k-3)y(k-1) + u(k-2)y(k-2) \\ &+ u(k-2)y(k-1) + u(k-1)y(k-2) + u(k-1)y(k-1). \end{aligned} \tag{4}$$

It is easily seen from (4) that, even for the simple case of a SISO system with $M = 3$ and $M_y = 2$, and $r = 2$, the model has twenty regressors. The poor scaling of the number of regressors with M, M_y, and r is one of the reasons why such models can result in identifiability issues and overfitting. Additionally, the regressors can be strongly intercorrelated, implying that a sparse model that groups the correlated regressors might be preferred to avoid overfitting.

Machine learning and especially algorithms that allow for variable selection, grouping of correlated variables, and sparsification provide a promising framework to address these limitations. Ridge regression (Hoerl and Kennard, 1988) is a penalization regression technique which imposes an L_2-norm bound on the regression coefficients w

$$\min_w \frac{1}{2N}\sum_{i=1}^{N}(y_i - x_i^\top w)^2 \quad \text{s.t.} \sum_{j=1}^{p} w_j^2 \le c, \tag{5}$$

where the y_i are the observations, the x_i are the regressors (or predictor variables), N is the number of observations, p is the number of the regressors, and c is a tuning parameter. The hyperparameter c is identified by cross-validation. Ridge regression is a continuous shrinkage method, and since it always keeps all the predictors in the model, a parsimonious model cannot be obtained.

The lasso is a method that imposes an L_1-penalty on the regression coefficients (Tibshirani, 1996), leading to both continuous shrinkage and variable selection thus resulting in a sparse model

$$\min_w \frac{1}{2N}\sum_{i=1}^{N}(y_i - x_i^\top w)^2 \quad \text{s.t.} \sum_{j=1}^{p} |w_j| \le c. \tag{6}$$

Cross-validation for identifying the hyperparameter c is needed. The lasso is ineffective for the $p \gg N$ case, because lasso can select at most N variables out of p regressors (Efron et al., 2004).

Another regularization method, EN,

$$\min_w \frac{1}{2N}\sum_{i=1}^{N}(y_i - x_i^\top w)^2 \quad \text{s.t.} \sum_{j=1}^{p} \frac{1-\alpha}{2} w_j^2 + \alpha|w_j| \le c, \ \alpha \in (0, 1], \tag{7}$$

produces models that are less sensitive to perturbations in the training data, while having the same or better performance than lasso (Zou and Hastie, 2005). EN selects groups of correlated variables to include them in the model, when lasso would select one variable from the group (Zou and Hastie, 2005). Constructing a model using EN involves two-dimensional cross-validation to identify the hyperparameters α and c. The EN approaches ridge regression for $\alpha \to 0$, and EN becomes lasso for $\alpha = 1$. EN performs well for models with $p \gg N$, which arises for example when a polynomial NARX model with large M, M_y, and r is considered, and results in sparse models.

Although the focus of this work is on building polynomial NARX models, the proposed approach can be implemented on other types of models where a large number of regressors arises, such as NARMAX or Volterra series models.

3. Sparse model identification methodology

The main criteria for predictive model selection that are adopted in this work are (Sun and Braatz, 2021; Efron et al., 2004; Zou and Hastie, 2005):

- high future prediction accuracy, evaluated through the root-mean-square error (RMSE) and the residual properties of the testing data,
- model interpretability and parsimony, which favor simpler models over models with very large number of regressors.

Data interrogation prior to model identification sheds light into the characteristics of the data and informs the selection of a model candidate pool. The degree of nonlinearity of the input–output data, multicollinearity of inputs (Sun and Braatz, 2021), and the autoregressive order of the process outputs are assessed during the data interrogation phase.

The Pearson's correlation coefficient is a measure of data pairwise linear relationships. It can take values between −1 and 1, with higher absolute values indicating a stronger linear relationship between the input and output. However, it assumes that the observations are independent, which does not hold for time series data. For time series data, the maximum absolute values of the cross-correlation of each IO pair can be used as an indicator of the nonlinearity of the data. Nonlinearity in the polynomial NARX models was considered by introducing model candidates with polynomial orders $r > 1$. The variable inflation factor (James et al., 2013) is calculated to assess multicollinearity when more than one input is present. Lastly, the partial autocorrelation function (PACF) of the output plotted versus the lags can inform the order of the autoregressive model. The lag at which the PACF becomes smaller than the confidence intervals indicates the order of the autoregressive terms that should be included in the NARX model. For example, if the PACF cuts off after the third lag, a model $M_y = 3$ would be considered. More details on statistical tests suitable for in-depth data interrogation are discussed in Sun and Braatz (2021).

During the model candidate design phase, the autoregressive order M_y, which is informed by the data interrogation, was fixed while various input horizons (M) and polynomial (r) orders were included in the model candidate pool based on domain knowledge. The final decision on M and r was made by evaluating the resulting model characteristics. This strategy of selecting the "best" model among a small set of models pre-selected based on the type of data and the application has been shown to produce accurate models for all noise levels (Sun and Braatz, 2021).

Due to the nature of time series data, splitting the available data into training, validation, and testing data must be done carefully. In this work, the testing data were collected independently of the training and validation data sets, and amounted to 25% of the total available data. The validation data set contained 33% of the remaining data. The training data preceded temporally the validation data. If only one data set is available, the validation data should also precede the testing data (Sun and Braatz, 2021). Implementing k-fold cross-validation with time series data requires additional attention. The data of the ith fold of a k-fold cross-validation need to be a subset of the data used in the $(i+1)$th fold and precede them temporally. Here, 3-fold cross-validation was implemented and in each fold about 33% of the available data was set aside for validation. The kth fold contained all available training and validation data points which were 75% of the available data.

The EN solution was returned for each value of the hyperparameter $a = \{0.1, 0.5, 0.7, 0.9, 0.95, 0.99\}$ in (7), for various values of c determined by the algorithm. For each fold of the 3-fold cross-validation, the training data were used to obtain the EN solution for various values of c and the RMSE was calculated on the validation data. The value of c corresponding to the minimum of the mean RMSE across all 3 folds was selected. This procedure was repeated for all values of a. Then the value of α was selected to give the smallest mean RMSE. Lastly, the entire training and validation data sets were used to retrain the model for the selected hyperparameters c and α. The testing data RMSE was calculated on the final model.

The training was implemented in MATLAB using the EN algorithm of the glmnet package (Friedman et al., 2010; Qian et al., 2013) on a workstation with Intel Xeon Gold 6152 CPU at 2.1 GHz and 256 GB RAM. The computational time associated with training all 17 model candidates of each case study was always below 10 min.

4. Nonlinear model predictive control formulation

The NMPC algorithm determines the control inputs by repeatedly solving an online optimization based on a nonlinear process model. A widely used MPC formulation is

$$\begin{aligned} \min_{\Delta \mathbf{u}} \quad & W_y \sum_{i=1}^{p} (y_{\text{ref}}(k+i) - \hat{y}(k+i))^2 + W_u \sum_{i=0}^{c-1} \Delta u(k+i)^2 \\ \text{s.t.} \quad & \mathbf{y}_{\min} \le \mathbf{y} \le \mathbf{y}_{\max}, \\ & \mathbf{u}_{\min} \le \mathbf{u} \le \mathbf{u}_{\max}, \\ & \Delta \mathbf{u}_{\min} \le \Delta \mathbf{u} \le \Delta \mathbf{u}_{\max}, \end{aligned} \tag{8}$$

where k is the sampling instant, $y_{\text{ref}}(k)$ is a reference (setpoint) of the plant output at instant k, $\hat{y}(k)$ is a prediction of the future plant output at instant k, $u(k)$ is the plant input at instant k, and $\Delta u(k) := u(k) - u(k-1)$ is the control move at instant k, p is the prediction horizon, c is the control horizon, W_y is a weighting factor on the outputs, W_u is a weighting factor on the inputs, $\mathbf{y}$ is a sequence of future outputs (controlled variables), $\mathbf{u}$ is a sequence of current and future inputs (manipulated variables), and $\Delta \mathbf{u}$ is a sequence of future control moves in the control horizon. In nonlinear MPC, a nonlinear model is used to obtain the predictions $\hat{y}(k+i)$.

The notation $x(i|k)$ stands for the value of variable x at instant i given the information up to (and including) the instant k. Multiple-step-ahead predictions for a NARX model with past horizon M and autoregressive order M_y can be obtained by writing out the individual predictions,

$$\begin{aligned} y(k|k) &= f(u(k-M), \dots, u(k-1), y(k-M_y), \dots, y(k-1)), \\ y(k+1|k) &= f(u(k-M+1), \dots, u(k), y(k-M_y+1), \dots, y(k)), \\ y(k+2|k) &= f(u(k-M+2), \dots, u(k+1), y(k-M_y+2), \dots, \hat{y}(k+1|k)), \\ &\vdots \\ y(k+p|k) &= f(u(k-M+p), \dots, u(k+p-1), y(k-M_y+p), \dots, \\ &\qquad \hat{y}(k+p-1|k)), \end{aligned} \tag{9}$$

where $u(k-M), \dots, u(k-1)$ are past inputs, $u(k)$ is the current input (determined by the optimization), $u(k+1), \dots, u(k+p-1)$ are future inputs, $y(k-M_y), \dots, y(k-1)$ are past outputs, $y(k)$ is the current output, and $\hat{y}(k+i|k)$, i positive integer, is a future output prediction defined below. The nonlinear function f is a linear combination of all monomials and polynomials of at most order r, which is the polynomial order of the polynomial NARX model. The prediction is propagated through for the entire prediction horizon p with $u(k+1) = u(k)$, $k = c, c+1, \dots, p-1$.

An integrator is included in the NMPC in the same fashion as in DMC (Cutler and Ramaker, 1979; Garcia and Morshedi, 1986; Ikonen, 2017). An unmeasured disturbance at k, which is the difference of the measured output from the predicted output $w(k|k) = y(k) - \hat{y}(k) = y(k) - y(k|k)$, is propagated throughout the entire prediction horizon such that $w(k+1|k) = \cdots = w(k+p|k) = w(k|k)$. Then the prediction of the future output becomes

$$\begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \vdots \\ \hat{y}(k+p|k) \end{bmatrix} = \begin{bmatrix} y(k+1|k) \\ y(k+2|k) \\ \vdots \\ y(k+p|k) \end{bmatrix} + y(k) - y(k|k). \tag{10}$$

Not all output measurements are available for the prediction of $y(k+2|k)$ and onward, so $\hat{y}(k+1|k)$, $\hat{y}(k+2|k), \dots, \hat{y}(k+p-1|k)$ are used in place of $y(k+1)$, $y(k+2), \dots, y(k+p-1)$. An estimator such as a Kalman filter can alternatively be used to estimate the outputs.

A first-order filter is implemented in the NMPC to address noise in the measurements,

$$y_{\text{filt}}(k) = \alpha y(k) + (1-\alpha) y_{\text{filt}}(k-1), \; y_{\text{filt}}(1) = y(1). \tag{11}$$

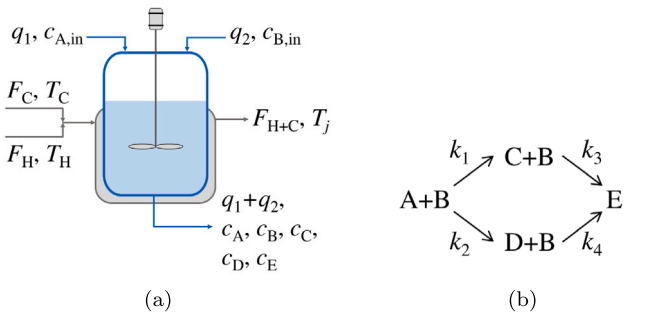


Fig. 1. Process flow diagram and multistep reaction network for the MIMO dynamic CSTR process.

The filtered values of the output instead of the measured values are provided to the controller when available.

This article follows standard NMPC tuning practice, e.g., (1) the prediction horizon was set just long enough to capture the dynamics, to avoid wasting online computation; (2) the control horizon was set just high enough to give good closed-loop performance, to avoid wasting online computation; and (3) the relative values of input and output weights chosen to weigh the relative values of the inputs and outputs.

The NMPC simulations were implemented in Julia (Bezanson et al., 2017). The nonlinear program arising in each iteration of the NMPC was solved by the solver NLopt (Johnson, 2020) using the JuMP algebraic modeling language (Dunning et al., 2017). JuMP employs automatic differentiation for the computation of gradients. The globally convergent method-of-moving-asymptotes (MMA) algorithm for gradient-based local optimization was used (Svanberg, 2002). Only input box constraints were implemented in the case studies presented here, but output and control move constraints can be included as needed, without any changes in the methodology. The nonlinear simulation of the process used in place of the "true" plant consisting of an ODE system was solved in the Differential Equations (Rackauckas and Nie, 2017) package in Julia. The NMPC computations were carried out in a laptop with Intel(R) Core(TM) i9-8950HK CPU at 2.90 GHz and 128 GB RAM.

5. Case study: Multiple-input multiple-output system

This case study considers a chemical reaction network taking place in a dynamic continuous stirred-tank reactor (CSTR) with two inputs and two outputs (see Fig. 1).

5.1. Process description

The multistep reaction network shown in Fig. 1(b) with second-order, bimolecular kinetics has been studied before for processes taking place in both a plug flow reactor (Reizman and Jensen, 2012; Shen and Braatz, 2016) and a dynamic CSTR (von Andrian and Braatz, 2020). Different reaction kinetics orders were assumed here which resulted in a more nonlinear input–output relationship. The process is run in a dynamic CSTR shown in Fig. 1(a). The concentration of the desired product species D in the outlet and the temperature are measured. The inlet flowrate of species A and the outlet flowrate can be manipulated, such that the reactor operates at constant volume. Species B is supplied at a constant flowrate and known concentration. Additionally, the reactor has a thermal jacket where heating and cooling fluids of known temperature are circulated. The total flow of the heating and cooling fluids is fixed but the ratio of the heating fluid can be manipulated.

The volumetric reaction rates are described by power law kinetics with an Arrhenius temperature dependence,

$$\begin{aligned} r_1 &= k_{0,1} \exp\left(\frac{-E_{A,1}}{RT}\right) c_A^{1/2} c_B, \quad r_2 = k_{0,2} \exp\left(\frac{-E_{A,2}}{RT}\right) c_A^{1/2} c_B, \\ r_3 &= k_{0,3} \exp\left(\frac{-E_{A,3}}{RT}\right) c_B^2 c_C, \quad r_4 = k_{0,4} \exp\left(\frac{-E_{A,4}}{RT}\right) c_B^2 c_D, \end{aligned} \tag{12}$$

Table 1
Parameter values for the process shown in Fig. 1.

Parameter	Description	Value	Units
$k_{0,1}$	Pre-exponential of reaction 1	$10^{3.4}$	L/(mol s)
$k_{0,2}$	Pre-exponential of reaction 2	$10^{3.5}$	L/(mol s)
$k_{0,3}$	Pre-exponential of reaction 3	$10^{4.9}$	L/(mol s)
$k_{0,4}$	Pre-exponential of reaction 4	10^{3}	L/(mol s)
$E_{A,1}$	Activation energy of reaction 1	27	kJ/mol
$E_{A,2}$	Activation energy of reaction 2	32.1	kJ/mol
$E_{A,3}$	Activation energy of reaction 3	60	kJ/mol
$E_{A,4}$	Activation energy of reaction 4	45	kJ/mol
q_2	Volumetric flowrate of stream supplying B	0.125	L/s
V	Reactor volume	1.25	L
$c_{\mathrm{A,in}}$	Feed concentration of species A	0.01	mol/L
$c_{\mathrm{B,in}}$	Feed concentration of species B	3.5	mol/L
U	Heat transfer coefficient	1100	W/(m^2 K)
A	Heat transfer area	0.1	m^2
ρ	Liquid mixture density	789.5	kg/m^3
ρ_j	Jacket fluid density	1110	kg/m^3
c_p	Liquid mixture heat capacity	2460	J/(kg K)
$c_{p,j}$	Jacket fluid heat capacity	2360	J/(kg K)
V_j	Jacket fluid volume	0.9375	L
T_{in}	Inlet stream temperature	298	K
T_{H}	Heating fluid temperature	473	K
T_{C}	Cooling fluid temperature	280	K
$F_{\mathrm{H+C}}$	Total heating and cooling fluid flowrate	4	L/s
ΔH_1	Enthalpy change of reaction 1	−10	kJ/mol
ΔH_2	Enthalpy change of reaction 2	−5	kJ/mol
ΔH_3	Enthalpy change of reaction 3	−20	kJ/mol
ΔH_4	Enthalpy change of reaction 4	−15.5	kJ/mol

where R is the ideal gas constant, T is the reactor absolute temperature, and c_i refers to the concentration of the ith species. The description and values of the kinetic parameters can be seen in Table 1.

The mass balances for each species within the reactor are described by

$$\begin{aligned}
\frac{\mathrm{d}c_{\mathrm{A}}}{\mathrm{d}t} &= \frac{q_1}{V}c_{\mathrm{A,in}} - \frac{q_1+q_2}{V}c_{\mathrm{A}} - r_1 - r_2,\\
\frac{\mathrm{d}c_{\mathrm{B}}}{\mathrm{d}t} &= \frac{q_2}{V}c_{\mathrm{B,in}} - \frac{q_1+q_2}{V}c_{\mathrm{B}} - r_1 - r_2 - r_3 - r_4,\\
\frac{\mathrm{d}c_{\mathrm{C}}}{\mathrm{d}t} &= -\frac{q_1+q_2}{V}c_{\mathrm{C}} - r_1 - r_3,\\
\frac{\mathrm{d}c_{\mathrm{D}}}{\mathrm{d}t} &= -\frac{q_1+q_2}{V}c_{\mathrm{D}} - r_2 - r_4,
\end{aligned} \tag{13}$$

where q_1 is the volumetric flowrate of the stream supplying species A to the reactor and V is the reactor volume. The values and descriptions of all other parameters are provided in Table 1.

The energy balances within the reactor and jacket are described by

$$\begin{aligned}
\frac{\mathrm{d}T}{\mathrm{d}t} &= \frac{UA}{\rho c_p V}(T_j - T) - \frac{\sum_i \Delta H_i r_i}{\rho c_p} + \frac{q_1+q_2}{V}(T_{\mathrm{in}} - T) \text{ and}\\
\frac{\mathrm{d}T_j}{\mathrm{d}t} &= \frac{UA}{\rho_j c_{p,j} V_j}(T - T_j) + \frac{1}{V_j}\left(r_{\mathrm{HC}}F_{\mathrm{H+C}}T_{\mathrm{H}} + (1-r_{\mathrm{HC}})F_{\mathrm{H+C}}T_{\mathrm{C}} - F_{\mathrm{H+C}}T_j\right)
\end{aligned} \tag{14}$$

respectively, where T_j is the jacket temperature, $r_{\mathrm{HC}} = F_{\mathrm{H}}/F_{\mathrm{H+C}}$, $F_{\mathrm{H+C}} = F_{\mathrm{H}} + F_{\mathrm{C}}$, F_{H} is the heating fluid flowrate, and F_{C} is the cooling fluid flowrate.

5.2. System identification

For the system identification, (12)–(14) were solved simultaneously, while sampling the process inputs $q_1 \sim \mathcal{U}(0, 0.625)$ and $r_{\mathrm{H+C}} \sim \mathcal{U}(0, 1)$. The input signal was repeated for 50 sampling instants. The measured values of the process outputs c_{D} and T were recorded every 2 s of operation. The sampling was selected to be five times faster than the largest process time scale which is about 10 s. Normal random noise acted on the true values resulting in the "measurements". The concentration measurement is $c_{\mathrm{D},m} = c_{\mathrm{D}} + \epsilon_{c_{\mathrm{D}}}$, $\epsilon_{c_{\mathrm{D}}} \sim \mathcal{N}(0, \sigma^2_{c_{\mathrm{D}}})$ and the temperature measurement is obtained as $T_m = T + \epsilon_T$, $\epsilon_T \sim \mathcal{N}(0, \sigma^2_T)$. Two noise levels were considered for the concentration measurement; a low level of noise with $\sigma_{c_{\mathrm{D}}} = 1.2 \times 10^{-5}$ mol/L and a medium level of noise with $\sigma_{c_{\mathrm{D}}} = 2.4 \times 10^{-5}$ mol/L. With the true value of c_{D} being $2\text{–}10 \times 10^{-4}$ mol/L, these noise levels correspond to about 6% and 12% maximum additive noise respectively. Higher levels of measurement noise for the concentration would imply a poorly designed process. The additive noise acting on the true value of the reactor temperature was set to have $\sigma_T = 0.17$ K which implies a maximum level of temperature noise of about 0.51 K.

5.3. Model construction for concentration prediction

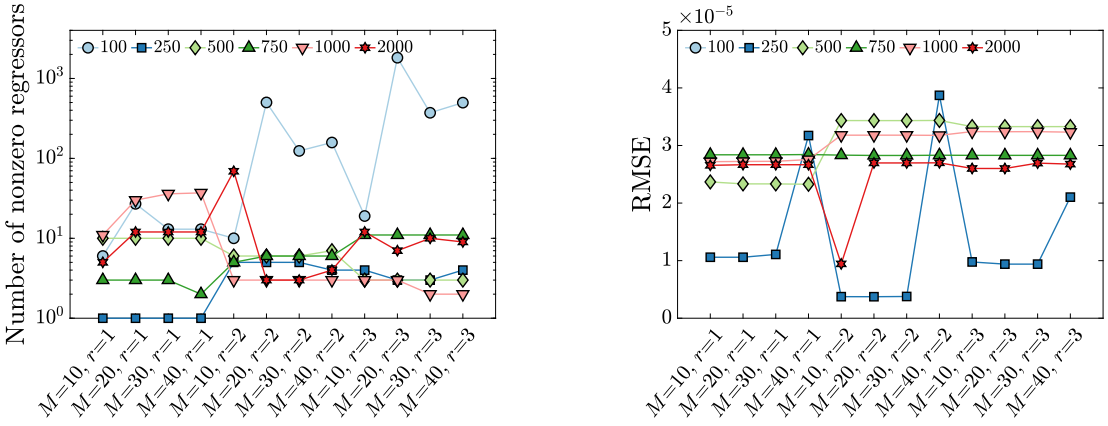
Both concentration and temperature models were constructed using data sets of various sizes and signal/noise ratios. The maximum value of the cross-correlation for the input–output pair c_{D}–q_1 is 0.26 and for c_{D}–$r_{\mathrm{H+C}}$ it is 0.32, implying that a nonlinear model for concentration might be more suitable. The maximum value of the cross-correlation for the input–output pair T–q_1 is −0.38 and for T–$r_{\mathrm{H+C}}$ is 0.80, implying that a linear model for temperature prediction might be sufficient. Model candidates with $M = \{10, 20, 30, 40\}$, $r = \{1, 2, 3\}$, and $M_y = 4$ were considered. Data sets with 100, 250, 500, 750, 1000, and 2000 data points were split into 75% training data and 25% testing data and separately trained. A 3-fold cross-validation for time series was also employed. The inputs u in the NARX model were q_1 and r_{HC} and the outputs y were c_{D} and T.

First, models for concentration were constructed with noise-free data. Sparse models were obtained using EN; most models have less than 100 regressors with a nonzero coefficient (Fig. 2(a)). To illustrate the level of sparsity, for the model candidate $M = 40$, $r = 3$ identified with 2000 data points, only 9 out of a total of 121,484 regressors were retained with a nonzero coefficient in the final model, indicating a sparsity of 99.99%.

The RMSE calculated for the testing data, is below 2% of the minimum concentration measurement value, except for the models built with the data set with 100 data points. The RMSE for models built with the 100 point data set is much larger compared to the RMSE of models built with the other data sets and is hence not depicted in Fig. 2(b). The smallest RMSE across all model candidates is observed for the models constructed with 250 and 2000 data points. Using either of those two data sets, the most parsimonious model candidate with the minimum RMSE is $M = 10$, $r = 2$. The test residual distributions for all model candidates increasingly approximate a normal distribution for larger data set sizes (SI Fig. 1). For models constructed with larger data sets, the residuals are smaller, as expected, since the higher number of high-quality data points increases the capacity to construct accurate models.

The test error autocorrelation plots for model candidates of second order constructed with 250, 500, and 2000 data points can be seen in Fig. 3. The autocorrelation vanishes within 20 lags for all models with $r = 2$, indicating that the dynamics are sufficiently captured by second-order polynomial models with a horizon of $M = 10$ or more. The model candidate with the minimum RMSE ($M = 10$, $r = 2$) constructed using data sets with 250, 500 or 2000 data points, sufficiently captures the process dynamics since the autocorrelation disappears at most after 10 lags.

Fig. 4 shows time-domain prediction, x–y plot, and testing residual characteristics for the model with $M = 10$, $r = 2$. For the models constructed with 2000 data points, the time-domain, x–y plot, and residual characteristics indicate that the model makes satisfactory predictions. On the other hand, the models constructed with 250 and 500 data points give skewed predictions. Worse predictions are observed for models with larger horizons (i.e., $M = 20$, $r = 2$, which can be seen in SI Fig. 4) or a polynomial order 3 (i.e., $M = 10$, $r = 3$ in SI Fig. 6). For a first-order model ($M = 10$, $r = 1$) the model is underpredicting



(a) Number of regressors with nonzero coefficients retained in the predictive model.

(b) RMSE evaluated on the testing data.

Fig. 2. Characteristics of constructed models with data sets of varying sizes. The legend indicates the total number of training and testing data points used. All models have an autoregressive order $M_y = 4$.

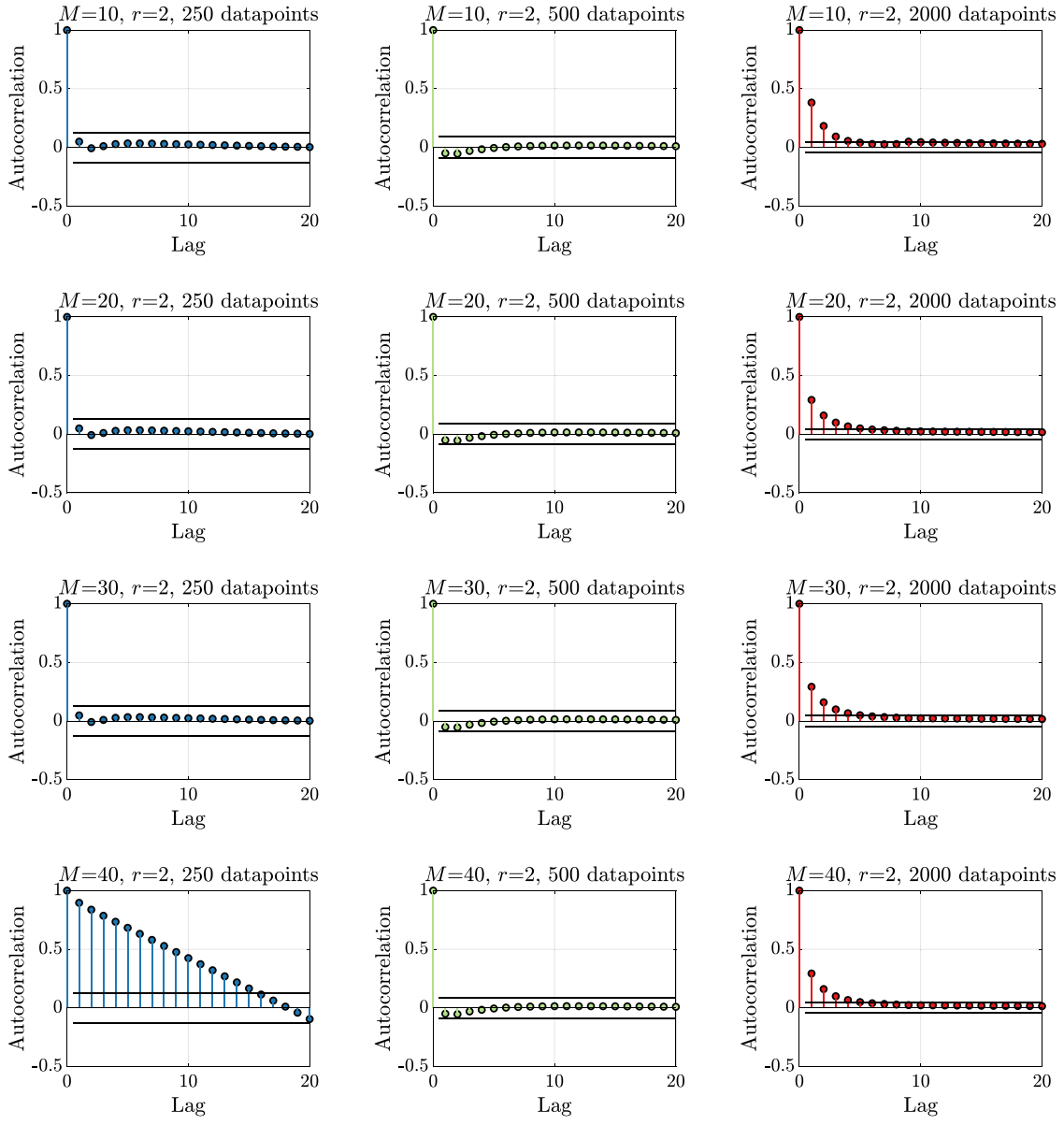


Fig. 3. c_D prediction residual autocorrelation analysis for the testing data for second-order models and various data set sizes used for training and testing.

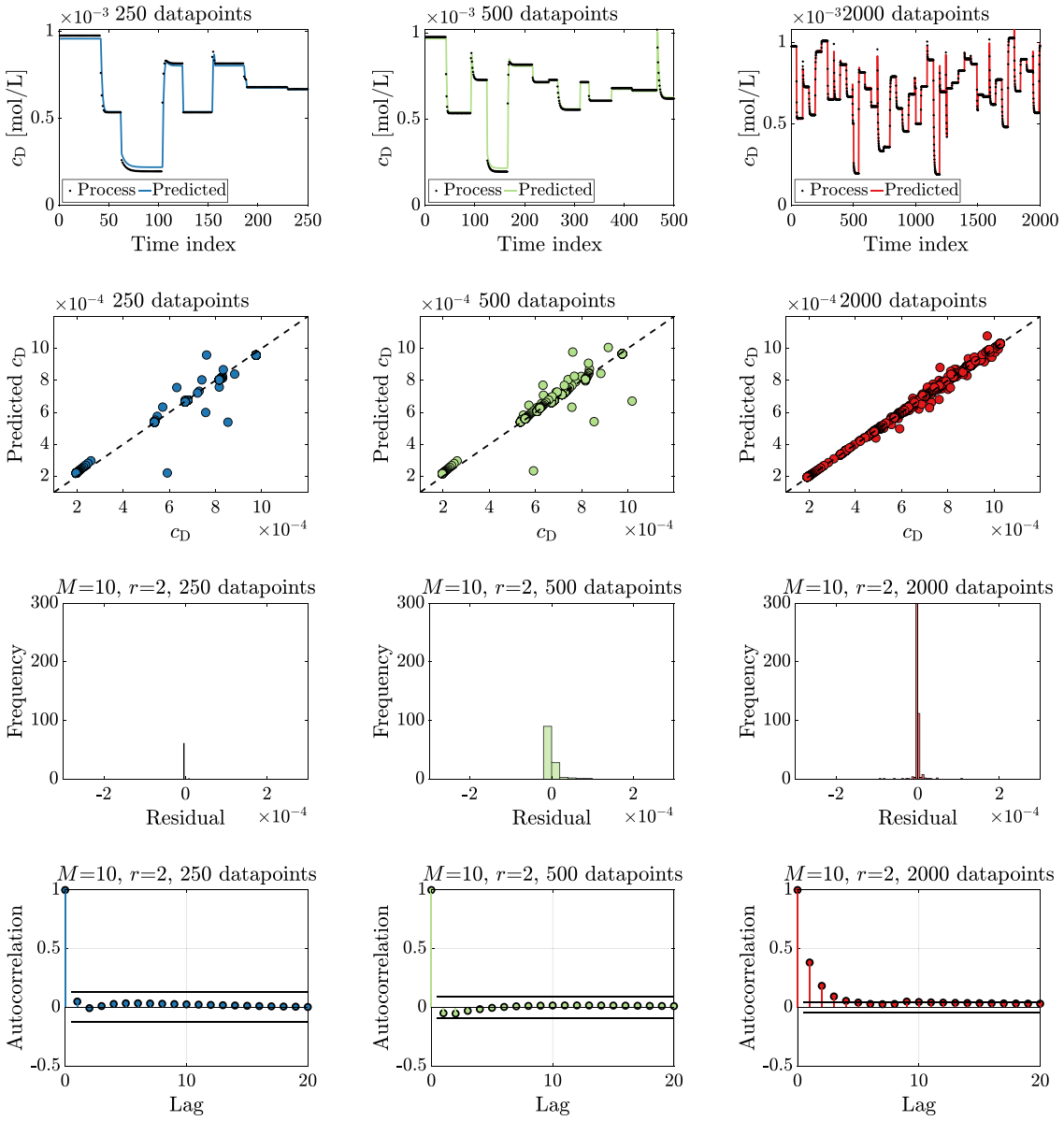


Fig. 4. c_D prediction performance metrics for the model $M = 10$, $r = 2$, for various data sets. The time-domain and x–y plots contain training and testing data points while the residual distribution and autocorrelation analyses correspond to the testing data residuals.

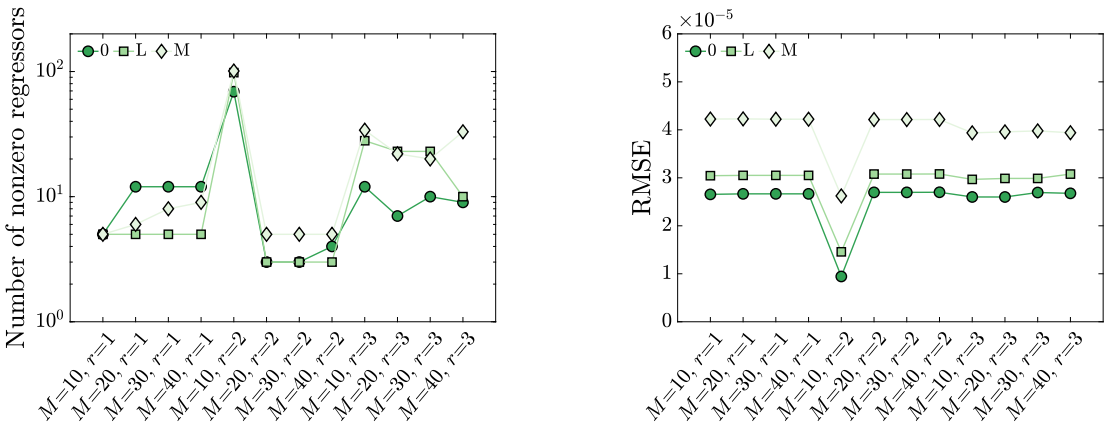
in the med- and high-range concentrations (see SI Fig. 5). In summary, to build a good prediction model for concentration, 2000 high-quality noise-free data points were needed for training and testing.

Since the data quality can be compromised by measurement noise, the ability to construct predictive models using data points affected by three different noise levels was assessed. The noise levels were denoted as "0" for perfect measurements of c_D, "L" for the low noise and "M" for the medium noise level, as defined in the System Identification section.

The data set with 2000 points was used to study the effect of measurement noise on model construction for concentration predictions. The RMSE of the models identified using the noisy data (Fig. 5) is about twice the standard deviation of the white noise affecting the data, where $\sigma_{c_D} = 1.2 \times 10^{-5}$ mol/L for "L" and $\sigma_{c_D} = 2.4 \times 10^{-5}$ mol/L for "M". The model with $M = 10$, $r = 2$ is quite sparse, although it is the least sparse compared to all other models in Fig. 5, and has the smallest RMSE across all levels of measurement noise. Additionally, its testing residuals are distributed quite normally around zero and over a smaller range compared to the residual distributions of all other model candidates (SI Fig. 7) for all levels of measurement noise. The good predictive capabilities of the model $M = 10$, $r = 2$ can be further verified by the time-domain, x–y, and test error autocorrelation plots in Fig. 6. Time-domain and testing residual plots for other models are appended for comparison purposes (SI Figs. 8–10). In conclusion, for reasonable levels of measurement noise, given enough high-quality data points, a good prediction model for c_D can be constructed.

5.4. Model construction for temperature prediction

A predictive model for temperature was constructed independently from the concentration model. Model candidates with $M = \{10, 20, 30, 40\}$ and $r = \{1, 2, 3\}$ were considered. The autoreregressive order for the outputs was $M_y = 4$ for all model candidates. Data sets with 100, 250, 500, 750, 1000, and 2000 data points were used to train the models. The model sparsity and corresponding testing RMSE for various model candidates are seen in Fig. 7. RMSE information on models constructed with 100 data points has been omitted from the plot due to the very large RMSE values of these models. It is apparent that 100 data points are not enough to describe the system behavior with a sparse



(a) Number of regressors with nonzero values used in the predictive model.

(b) RMSE evaluated on the testing data.

Fig. 5. Characteristics of concentration predictive models constructed with data sets of 2000 data points. The legend indicates the level of noise impacting the measurements as "0" (no noise), "L" (low level of noise) and "M" (medium level of noise). All models have an order $M_y = 4$.

model and the resulting models are very inaccurate. The sparsity for the resulting temperature models is on the same order of magnitude as the sparsity for the concentration models in most cases. The values of the testing RMSE (Fig. 7(b)) are below 3 degrees for most models, implying that the constructed models are quite accurate and the predictive differences between them are not significant. The smallest values for the testing RMSE are obtained for models of polynomial order $r = 2$ for the data set with 1000 data points and for $r = 1$ for the largest data set (2000 data points). However, the smallest RMSE does not necessarily point to the best model; a parsimonious model should be favored when the model differences are small. In this case, first-order models seem to have a small enough RMSE compared to second-order models. Therefore, first-order models are most likely sufficient for a temperature predictive model. This hypothesis is further assessed by examining the test residual properties.

The test residual distributions increasingly approximate a normal distribution as the size of the data set used to construct the models increases (SI Fig. 11), indicating that training models with more high-quality data points results in less skewed predictions. Additionally, the variance of the test residuals of the first-order models is smaller than the variance of the residuals of higher-order models.

The test residual autocorrelation vanishes rapidly for first-order polynomial models constructed with 2000 data points but not for first-order models constructed with fewer data points (Fig. 8). The autocorrelation for second- and third-order polynomial models (SI Figs. 12 and 13) does not reduce as fast as the autocorrelation for models with $r = 1$. For first-order models, the model horizon M does not significantly impact the autocorrelation function for the models constructed with 2000 data points or more, implying that $M = 10$, $r = 1$ would be sufficient to capture the system dynamics.

Time-domain prediction and x–y plots as well as the corresponding testing residual results for the model $M = 10$, $r = 1$ are shown in Fig. 9. The model constructed with more data points has better predictive capabilities while models constructed with 250 or 500 data points exhibit skewed predictions. Results for other models are shown in SI Figs. 14–17. All these plots indicate that a wider model horizon is not necessary and that higher-order models do not offer any advantage to a first-order model. Therefore, the simpler models are favored during the final model selection process, due to their ability to capture the process behavior adequately while maintaining simplicity of representation.

Predictive models for temperature were also constructed using 2000 data points affected by measurement noise. The notation "0", "L", and "M" corresponds to no, low, and medium noise levels impacting the measurement of c_D while the level of noise impacting the temperature measurement was constant. Noisier data used for temperature modeling resulted in sparser models in most cases (Fig. 10(a)). The testing RMSE for the models constructed with noisy data was about 10% higher than the RMSE for the models constructed with noise-free data (Fig. 10(b)). The RMSE was about 0.1% of the temperature value, suggesting that even under the presence of measurement noise the resulting model predictions are quite accurate when enough data points are used to train the models. The minimum RMSE for the models constructed with noisy data was observed for first-order models, in agreement with the observation for the noise-free data case. The testing residual distributions (see SI Fig. 18) were consistent with previous observations (see SI Fig. 7), that increasing levels of measurement noise result in residual distributions that increasingly approximate the normal distribution.

Analyzing the models trained with the noisy data pointed towards selecting the same final model as when analyzing models trained with noise-free data, with $M = 10$, $r = 1$ (see Fig. 11). Time-domain and testing residual plots for other models are appended for comparison purposes (SI Figs. 19–22). The small differences between the various models point to the final selection of the simplest model. Therefore, the model selected for temperature prediction was $M = 10$, $r = 1$.

5.5. Nonlinear model predictive control

The continuous stirred-tank reactor was controlled using NMPC. The model with input horizon $M = 10$, autoregressive order $M_y = 4$, and polynomial order $r = 2$ was selected for the concentration prediction and the model with $M = 10$, $M_y = 4$, and $r = 1$ was selected for the prediction of temperature. Models constructed with 2000 data points with noise-free ("0") and noisy ("M") data were used in closed-loop simulations (Figs. 12(a)–12(f)). The first-principles model was used in place of the "true" plant and the ODE system was solved in Julia (Rackauckas and Nie, 2017) with the solver `Rosenbrock23` which is appropriate for stiff equations at high tolerances. The level of measurement noise which impacted the closed-loop operation was the same as the level of noise impacting the c_D and T measurements used for model construction. The controller calculated the future input sequence every 2 s which matches the sampling frequency of data collection during the system identification phase. The controller tuning parameters and listed in Table 2. Table 3 summarizes the disturbances acting on the process. The implemented control moves are shown in Figs. 12(e) and 12(f).

The reactor was initialized at some steady state different from the setpoint at $t = 0$. During the first 9 sampling instants of the operation, a model prediction was not available since at least 4 past outputs and 9 past inputs (and a guess for the current input) were required to predict the future outputs. Initial input values (here $q_1 = 0.3$

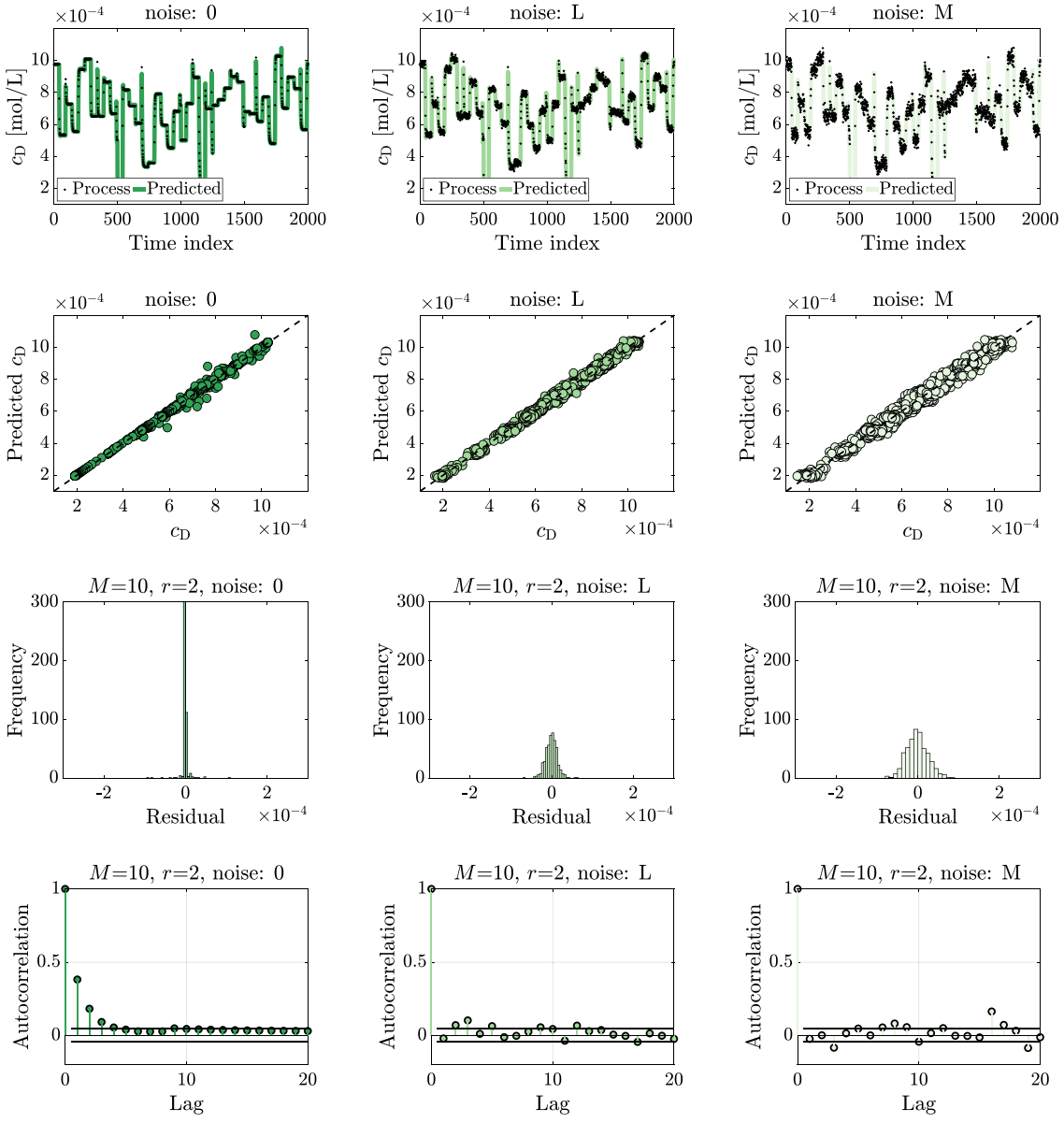


Fig. 6. c_D prediction performance metrics for the model $M = 10$, $r = 2$, constructed with 2000 data points, for various noise levels. The time-domain and x–y plots contain training and testing data points while the residual distribution and autocorrelation analyses correspond to the testing data residuals.

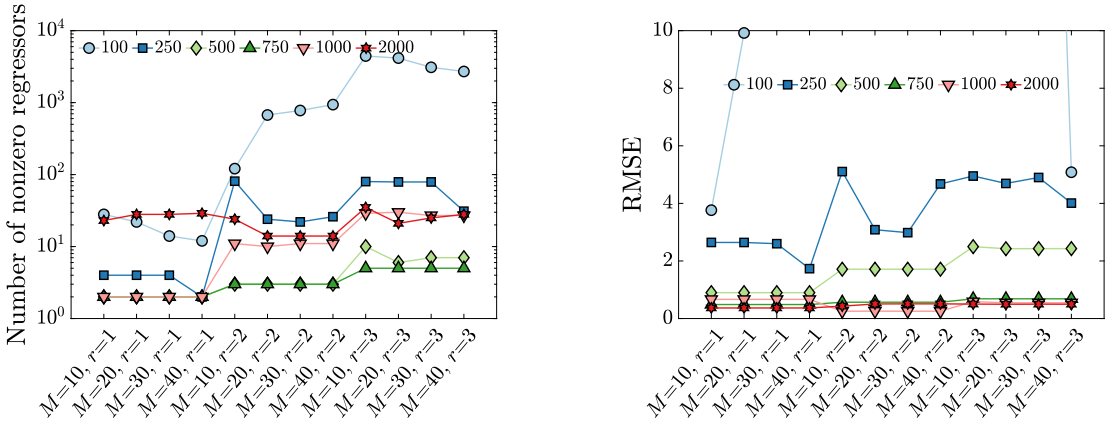
Table 2
NMPC tuning parameters for the MIMO case study.

Prediction horizon p	6
Control horizon c	3
Input q_1 weight W_u	1
Input r_{HC} weight W_u	2
Output c_D weight W_y	2×10^5
Output T weight W_y	4×10^{-4}
First-order filter α for c_D	0.4
First-order filter α for T	0.7

Table 3
Step disturbances acting on the closed-loop operation with responses shown in Fig. 12.

Disturbance	Magnitude	Start time [min]	End time [min]
q_2	+10%	2.5	3.5
T_C	+5 K	2.5	3.5
$c_{A,in}$	−10%	4.5	6.5
q_2	−10%	7.5	8.5
T_C	−5 K	7.5	8.5
$c_{A,in}$	+10%	9	11

L/s and $r_{HC} = 0.5$) were used before the controller started acting on the process. All controllers successfully rejected the disturbances listed in Table 3, usually within 0.5 min and at most within 1.3 min. Disturbances in the inlet concentration of species A, $c_{A,in}$, caused larger deviations from the setpoint compared to simultaneous disturbances in the inlet flowrate q_2 and cooling fluid temperature T_C. The plant would reach the setpoint at about 0.8 min after a setpoint change was introduced. This timescale is about 4.8 times the process time scale. The accuracy of the concentration predictive model was compromised when unmeasured disturbances were acting on the process and this was more prominent when measurement noise was also present. The loss of predictive accuracy is noticeable, for example, between the 4.5–6.5 min marks (Fig. 12(c)), when $c_{A,in}$ is reduced by 10%. The predictive model



(a) Number of regressors with nonzero coefficients retained in the predictive model.

(b) RMSE evaluated on the testing data.

Fig. 7. Characteristics of the reactor temperature prediction models. The models were constructed with data sets of varying sizes. The legend indicates the total number of training and testing data points used. All models have an autoregressive order $M_y = 4$.

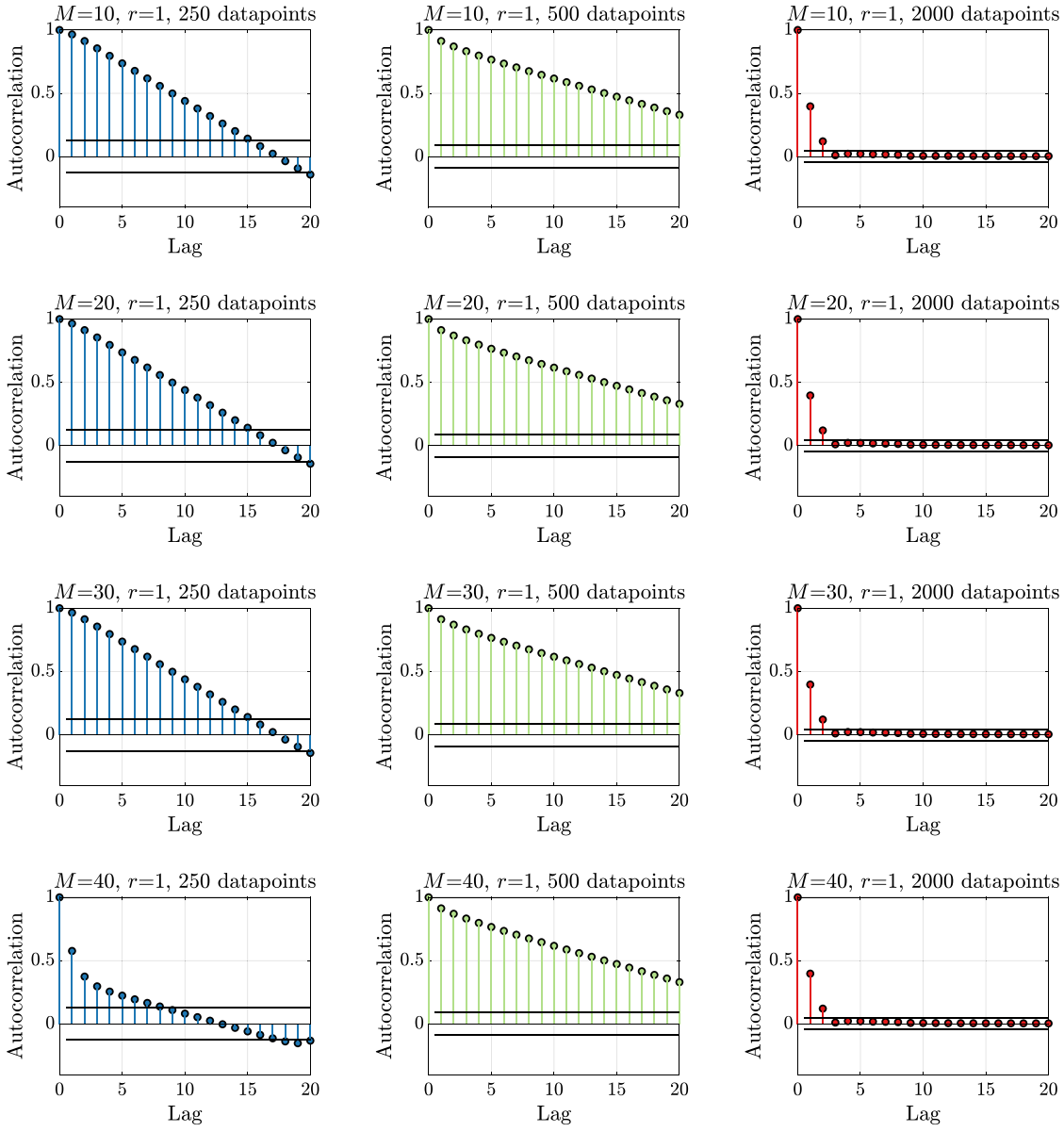


Fig. 8. Temperature prediction residual autocorrelation analysis for the testing data for first-order models and various data set sizes used for training and testing.

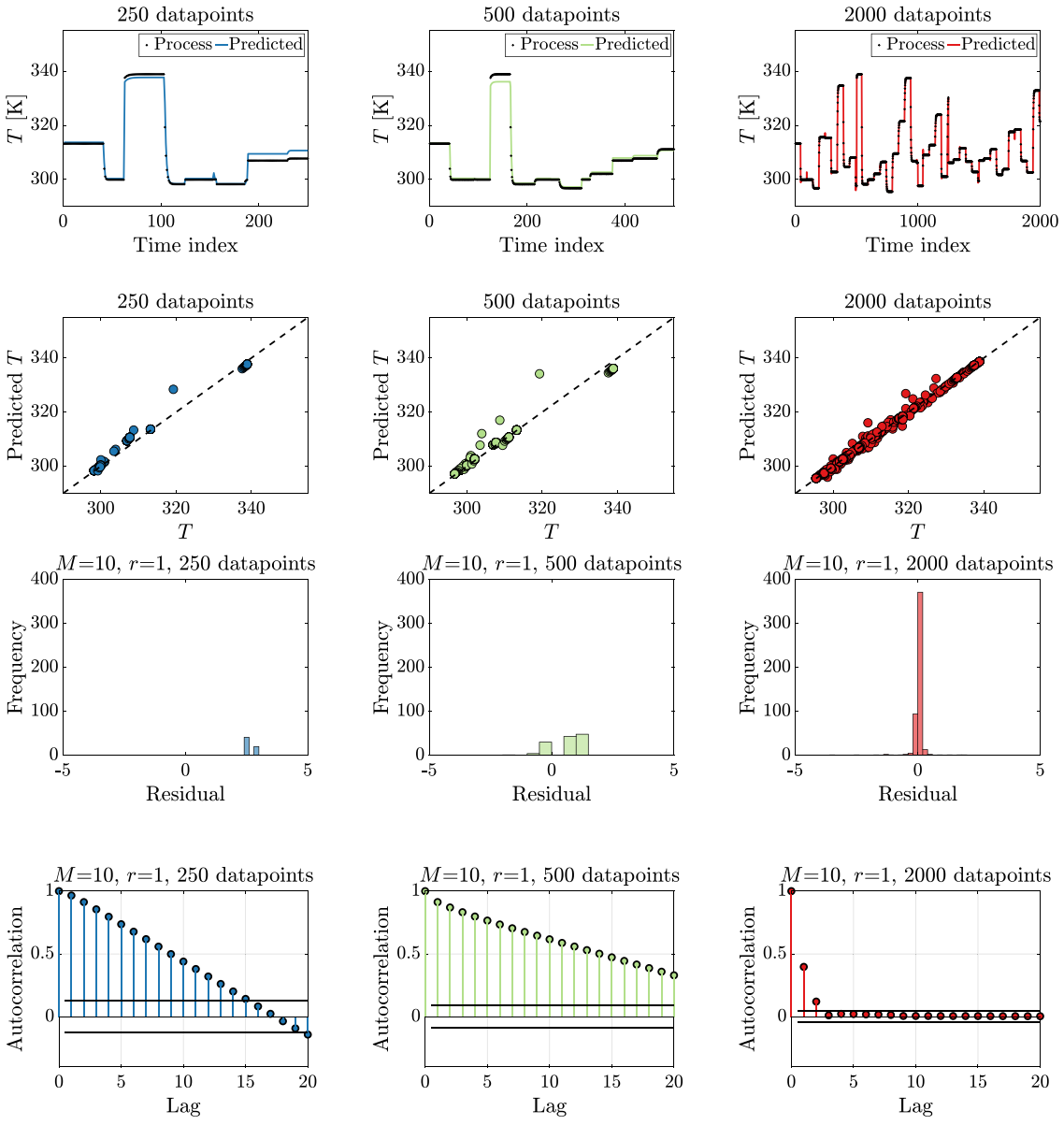


Fig. 9. Temperature prediction performance metrics for the model $M = 10$, $r = 1$, for various data sets. The time-domain and x–y plots contain training and testing data points while the residual distribution and autocorrelation analyses correspond to the testing data residuals.

for temperature is less susceptible to loss of predictive accuracy when noise and disturbances are present. Despite the loss of accuracy in the prediction, the controller successfully brought the plant to the setpoint due to integral action (see Figs. 12(a)–12(d)).

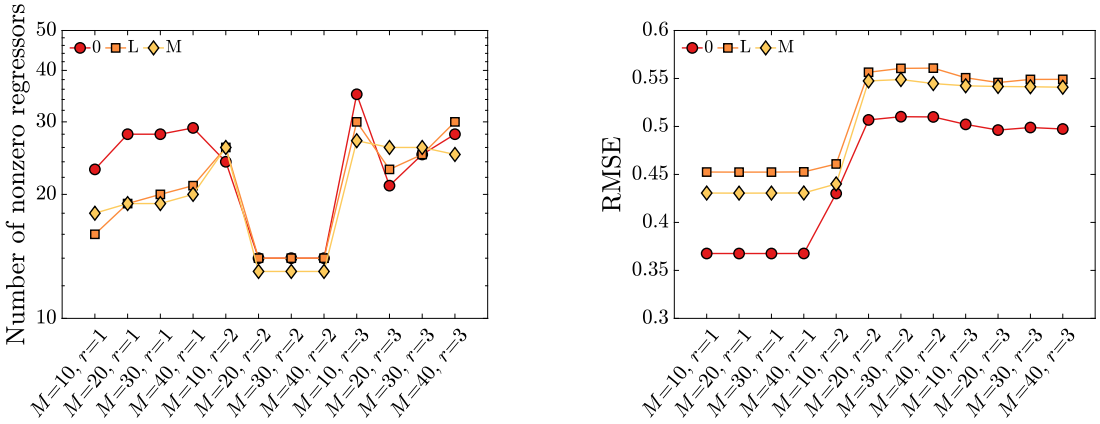
The nonlinear optimization was solved every 2 s of operation with an associated computational time shown in Fig. 13. Since the computational time is two orders of magnitude faster, the controller is real-time implementable. The computations were carried out in a laptop with Intel(R) Core(TM) i9-8950HK CPU at 2.90 GHz and 128 GB RAM.

Closed-loop responses for MPCs using different NARX models are shown in Fig. 14. The corresponding controller actions are shown in Fig. 15 and the associated computational times of the optimization are shown in Fig. 16. The NMPC described earlier (NMPC1) is compared to linear MPC (LMPC) where a NARX model with $M_y = 4$, $M = 10$, and $r = 1$ for both c_D and T predictions is implemented, as well as another NMPC scheme (NMPC2) with a NARX model with $M_y = 4$, $M = 10$, $r = 3$ for both c_D and T predictions. The closed-loop performance differences among the LMPC and the NMPC schemes are significant. LMPC cannot control concentration to its setpoint due to the model-plant mismatch. NMPC2 has slower dynamics compared to NMPC1. All MPCs were tuned as per Table 2.

6. Conclusion

A methodology is proposed for identifying sparse polynomial NARX models using elastic net for use in real-time implementable NMPC. Good closed-loop performance under setpoint changes, unmeasured disturbances, and measurement noise is demonstrated for a case study of a MIMO dynamic continuous stirred-tank reactor with a heated jacket.

First-principles simulations were used to generate representative data of the process dynamics. The MIMO case study system exhibited strong nonlinearities between the inputs and outputs. Data interrogation informed the autoregressive order of the polynomial model



(a) Number of regressors with nonzero values used in the predictive model.

(b) RMSE evaluated on the testing data.

Fig. 10. Characteristics of temperature prediction models, constructed using 2000 data points. The legend indicates the level of noise impacting the concentration measurements as "0" (no noise), "L" (low level of noise), and "M" (medium level of noise). All models have an order $M_y = 4$.

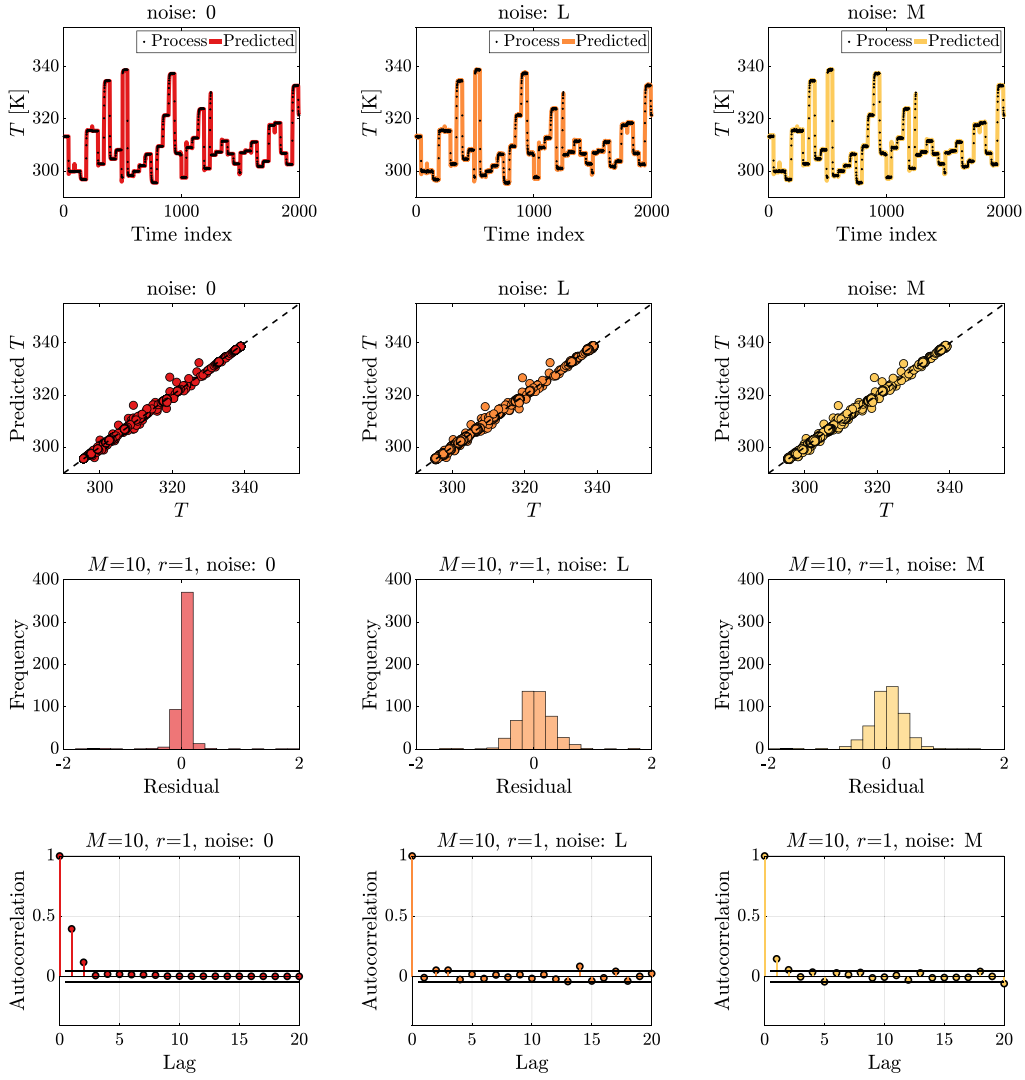
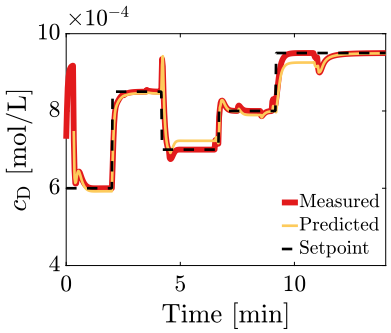
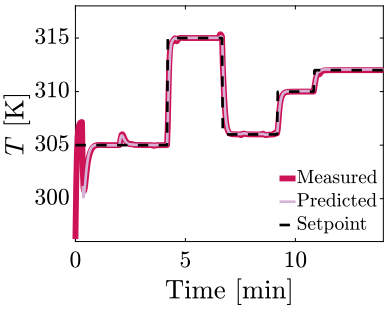


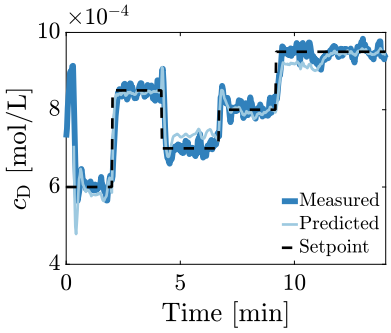
Fig. 11. Temperature prediction performance metrics for the model $M = 10$, $r = 1$, constructed with 2000 data points, for various noise levels. The time-domain and x–y plots contain training and testing data points while the residual distribution and autocorrelation analyses correspond to the testing data residuals.



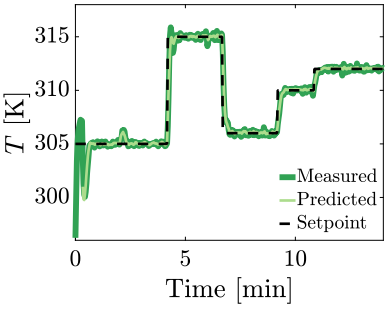
(a) Time evolution of c_D in closed loop. The measurements were noise-free.



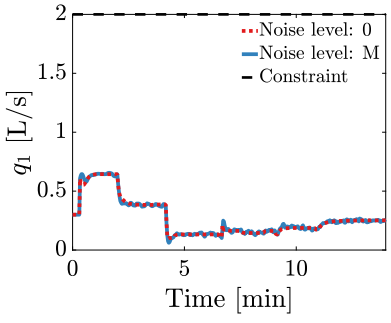
(b) Time evolution of T in closed loop. The measurements were noise-free.



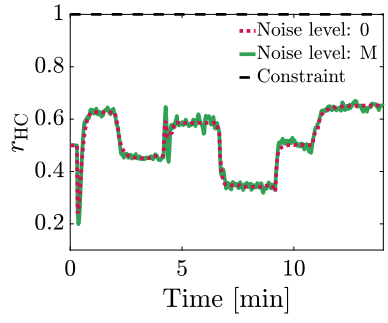
(c) Time evolution of c_D in closed loop. Medium noise level impacted the measurements.



(d) Time evolution of T in closed loop. Medium noise level impacted the measurements.



(e) Time evolution of the control input q_1.



(f) Time evolution of the control input r_{HC}.

Fig. 12. Closed-loop response of the concentration of species D, the reactor temperature and the associated manipulated variables during a setpoint tracking and unmeasured disturbance program.

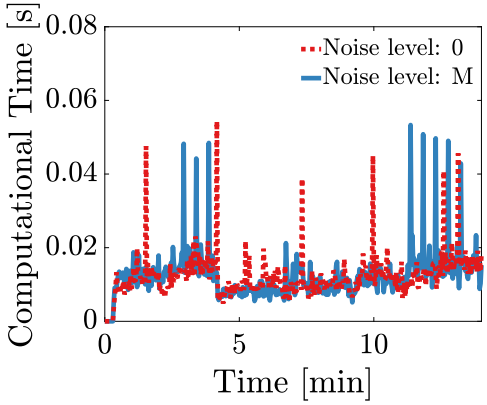


Fig. 13. Computational time associated with the NMPC controller with closed-loop responses shown in Fig. 12.

candidates, to limit the model candidate pool to a small set. The relationship between the process characteristics – such as extent of nonlinearity and the signal-to-noise ratio – and the quantity of data needed for model construction was examined. A linear and a nonlinear predictive model were identified for the two system outputs respectively. In both cases, the larger the data sets used for model training and cross-validation, the better the predictive accuracy of the model. The correct model structure was identified even under the presence of various noise levels, provided that a large enough data set was available.

The resulting NMPC algorithm is implementable in real time, with computational times about two orders of magnitude faster than the control sampling time. The sparse algebraic predictive models and the use of automatic differentiation, internally implemented in JuMP, significantly contribute to the fast solution of the optimization in each control iteration. Good, fast, reliable solvers are central in the implementation of NMPC, especially if such systems engineering solutions aim to become part of regulated software packages with potential for applications in manufacturing.

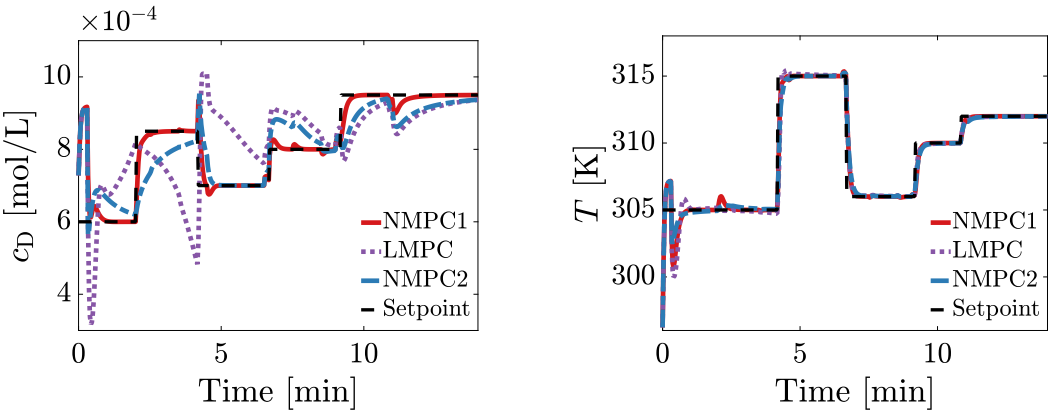


Fig. 14. Closed-loop response of the concentration of species D and the reactor temperature during a setpoint tracking and unmeasured disturbance program based on various NARX models.

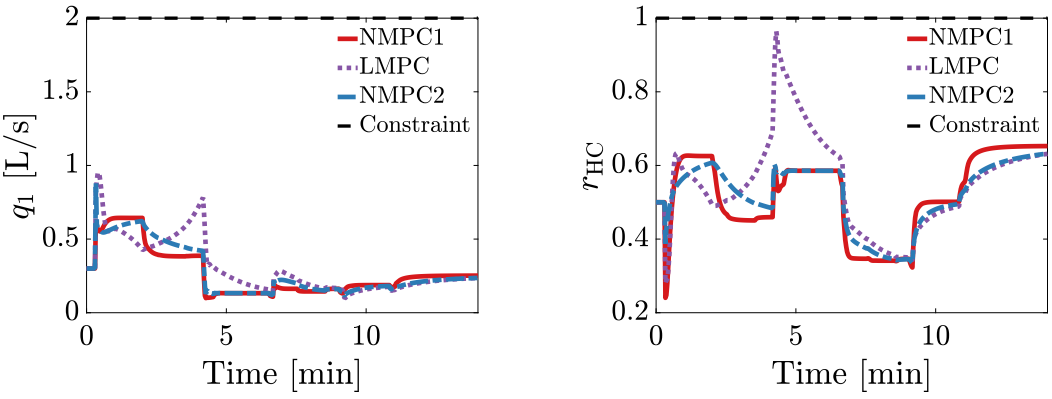


Fig. 15. Time evolution of the manipulated variables during a setpoint tracking and unmeasured disturbance program based on various NARX models.

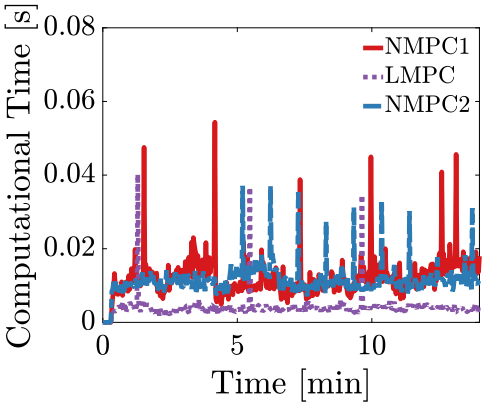


Fig. 16. Computational time associated with the NMPC controller with closed-loop responses shown in Fig. 14.

CRediT authorship contribution statement

Anastasia Nikolakopoulou: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Richard D. Braatz:** Conceptualization, Funding acquisition, Methodology, Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

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

Acknowledgments

This material is based upon work supported in part by the Defense Advanced Research Projects Agency (DARPA), United States Make-It program under contract No. ARO W911NF-16-2-0023. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA). Dr. Benoît Legat's help with Julia and JuMP implementation is also acknowledged.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.compchemeng.2023.108272.

References

Bai, L., Coca, D., 2007. NARMAX based nonlinear predictive control. In: Proc. of the European Control Conference. pp. 2319–2325.

Basu, S., Michailidis, G., 2015. Regularized estimation in sparse high-dimensional time series models. Ann. Statist. 43 (4), 1535–1567.

Baxendale, I.R., Braatz, R.D., Hodnett, B.K., Jensen, K.F., Johnson, M.D., Sharratt, P., Sherlock, J.-P., Florence, A.J., 2015. Achieving continuous manufacturing: Technologies and approaches for synthesis, workup and isolation of drug substance. J. Pharm. Sci. 104 (3), 781–791.

Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B., 2017. Julia: A fresh approach to numerical computing. SIAM Rev. 59 (1), 65–98. http://dx.doi.org/10.1137/141000671, URL https://epubs.siam.org/doi/10.1137/141000671.

Billings, S.A., 2013. Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains. Wiley, Chichester, UK.

Billings, S.A., Voon, W.S.F., 1986. A prediction-error and stepwise-regression estimation algorithm for non-linear systems. Internat. J. Control 44 (3), 803–822.

Calafiore, G.C., El Ghaoui, L.M., Novara, C., 2015. Sparse identification of posynomial models. Automatica 59, 27–34.

Calafiore, G.C., Novara, C., Taragna, M., 2017. Leading impulse response identification via the elastic net criterion. Automatica 80, 75–87.

Chiu, C.-C., Yao, Y., 2013. Multiway elastic net (MEN) for final product quality prediction and quality-related analysis of batch processes. Chemometr. Intell. Lab. Syst. 125, 153–165.

Cutler, C.R., Ramaker, B.L., 1979. Dynamic Matrix Control – a computer control algorithm. In: AIChE National Meeting. Houston, Texas.

De Nicolao, G., Magni, L., Scattolini, R., 1997. Stabilizing predictive control of nonlinear ARX models. Automatica 33 (9), 1691–1697.

Dunning, I., Huchette, J., Lubin, M., 2017. JuMP: A modeling language for mathematical optimization. SIAM Rev. 59 (2), 295–320. http://dx.doi.org/10.1137/15M1020575.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., 2004. Least angle regression. Ann. Statist. 32 (2), 407–499.

Friedman, J., Hastie, T., Tibshirani, R., 2010. Regularization paths for generalized linear models via coordinate descent. J. Stat. Softw. 33 (1), 1–22, URL https://www.jstatsoft.org/v33/i01/.

Garcia, C.E., Morshedi, A.M., 1986. Quadratic programming solution of dynamic matrix control (QDMC). Chem. Eng. Commun. 46 (1–3), 73–87.

Harinath, E., Foguth, L.C., Paulson, J.A., Braatz, R.D., 2019. Model predictive control of polynomial systems. In: Handbook of Model Predictive Control. Springer, Birkhäuser, Basel.

Henrion, D., Lasserre, J.B., 2002. GloptiPoly: Global optimization over polynomials with matlab and SeDuMi. In: Proc. of the IEEE Conference on Decision and Control. pp. 747–752.

Hoerl, A., Kennard, R., 1988. Ridge regression. In: Encyclopedia of Statistical Sciences, Vol. 8. Wiley, New York, pp. 129–136.

Huusom, J.K., Jørgensen, J.B., 2014. A realistic process example for MIMO MPC based on autoregressive models. In: Proc. IFAC World Congress. pp. 3086–3091.

Ikonen, E., 2017. Model Predictive Control and State Estimation. Technical Report, University of Oulu, Finland.

James, G., Witten, D., Hastie, T., Tibshirani, R., 2013. An Introduction to Statistical Learning. Springer, New York, NY.

Johnson, S.G., 2020. The NLopt nonlinear-optimization package. NLopt 2.7.0, http://github.com/stevengj/nlopt.

Kukreja, S.L., Löfberg, J., Brenner, M.J., 2006. A least absolute shrinkage and selection operator (LASSO) for nonlinear system identification. In: Proc. of the IFAC Symposium on System Identification. pp. 814–819.

Mascia, S., Heider, P.L., Zhang, H., Lakerveld, R., Benyahia, B., Barton, P.I., Braatz, R.D., Cooney, C.L., Evans, J.M.B., Jamison, T.F., Jensen, K.F., Myerson, A.S., Trout, B.L., 2013. End-to-end continuous manufacturing of pharmaceuticals: Integrated synthesis, purification, and final dosage formation. Angew. Chemie Int. Ed. 52 (47), 12359–12363.

MATLAB System Identification Toolbox, 2019. The MathWorks Inc., Natick, Massachusetts.

Myerson, A.S., Krumme, M., Nasr, M., Thomas, H., Braatz, R.D., 2015. Control systems engineering in continuous pharmaceutical manufacturing. J. Pharm. Sci. 104 (3), 832–839.

Nikolakopoulou, A., von Andrian, M., Braatz, R.D., 2019. Plantwide control of a compact modular reconfigurable system for continuous-flow pharmaceutical manufacturing. In: Proc. American Control Conference. pp. 2158–2163.

Nikolakopoulou, A., von Andrian, M., Braatz, R.D., 2020a. Fast model predictive control of startup of a compact modular reconfigurable system for continuous-flow pharmaceutical manufacturing. In: Proc. American Control Conference. pp. 2778–2783.

Nikolakopoulou, A., Braatz, R.D., 2022. Fast nonlinear model predictive control of distributed parameter systems. In: Proc. of the American Control Conference. pp. 994–999.

Nikolakopoulou, A., Hong, M.S., Braatz, R.D., 2020b. Feedback control of dynamic artificial neural networks using linear matrix inequalities. In: Proc. IEEE Conference on Decision and Control. pp. 2210–2215.

Nikolakopoulou, A., Hong, M.S., Braatz, R.D., 2021. Output feedback control and observer design for dynamic artificial neural networks. In: Proc. of the American Control Conference. pp. 2613–2618.

Pearson, R.K., 1995. Nonlinear input/output modelling. J. Process Control 5 (4), 197–211.

Pearson, R., 2003. Selecting nonlinear model structures for computer control. J. Process Control 13, 1–26.

Prajna, S., Papachristodoulou, A., Parrilo, P.A., 2002. Introducing SOSTOOLS: A general purpose sum of squares programming solver. In: Proc. of the IEEE Conference on Decision and Control. pp. 741–746.

Qian, J., Hastie, T., Friedman, J., Tibshirani, R., Simon, N., 2013. Glmnet for matlab. URL http://hastie.su.domains/glmnet_matlab/.

Qin, S.J., Badgwell, T.A., 2003. A survey of industrial model predictive control technology. Control Eng. Pract. 11, 733–764.

Rackauckas, C., Nie, Q., 2017. Differentialequations.jl– a performant and feature-rich ecosystem for solving differential equations in Julia. J. Open Res. Softw. 5 (1), 15.

Reizman, B.J., Jensen, K.F., 2012. An automated continuous-flow platform for the estimation of multistep reaction kinetics. Organic Process Res. Dev. 16, 1770–1782.

Ribeiro, A.H., Aguirre, L.A., 2018. Lasso regularization paths for NARMAX models via coordinate descent. In: Proc. of the American Control Conference. pp. 5268–5273.

Rogers, L., Briggs, N., Achermann, R., Adamo, A., Azad, M., Brancazio, D., Capellades, G., Hammersmith, G., Hart, T., Imbrogno, J., Kelly, L.P., Liang, G., Neurohr, C., Rapp, K., Russell, M.G., Salz, C., Thomas, D.A., Weimann, L., Jamison, T.F., Myerson, A.S., Jensen, K.F., 2020. Continuous production of five active pharmaceutical ingredients in flexible plug-and-play modules: A demonstration campaign. Organic Process Res. Dev. 24, 2183–2196.

Severson, K., VanAntwerp, J.G., Natarajan, V., Antoniou, C., Thömmes, J., Braatz, R.D., 2015. Elastic net with Monte Carlo sampling for data-based modeling in biopharmaceutical manufacturing facilities. Comput. Chem. Eng. 80, 30–36.

Shen, D.E., Braatz, R.D., 2016. Polynomial chaos-based robust design of systems with probabilistic uncertainties. AIChE J. 62 (9), 3310–3318.

Sriniwas, G.R., Arkun, Y., 1997. A global solution to the nonlinear model predictive control algorithms using polynomial ARX models. Comput. Chem. Eng. 21 (4), 431–439.

Sriniwas, G.R., Arkun, Y., Chien, I.-L., Ogunnaike, B.A., 1995. Nonlinear identification and control of a high-purity distillation column: A case study. J. Process Control 5 (3), 149–162.

Sun, W., Braatz, R.D., 2020. ALVEN: Algebraic learning via elastic net for static and dynamic nonlinear model identification. Comput. Chem. Eng. 143.

Sun, W., Braatz, R.D., 2021. Smart process analytics for predictive modeling. Comput. Chem. Eng. 144.

Svanberg, K., 2002. A class of globally convergent optimization methods based on conservative convex separable approximations. SIAM J. Optim. 12 (2), 555–573.

Tibshirani, R., 1996. Regression shrinkage and selection via the Lasso. J. R. Stat. Soc. B 58 (1), 267–288.

von Andrian, M., Braatz, R.D., 2020. Stochastic dynamic optimization and model predictive control based on polynomial chaos theory and symbolic arithmetic. In: Proc. of the American Control Conference. pp. 3399–3404.

von Stosch, M., Oliveira, R., Peres, J., de Azevedo, S.F., 2014. Hybrid semi-parametric modeling in process systems engineering: Past, present and future. Comput. Chem. Eng. 60, 86–101.

Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. J. R. Stat. Soc. B 67 (2), 301–320.