Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 55-20 (2022) 421–426

Neural ODEs and differential flatness for total least squares parameter estimation

Aike Aline Tappe * **Moritz Schulze** *,** **René Schenkendorf** ***

* *Institute of Energy and Process Systems Engineering, TU Braunschweig, Langer Kamp 19B, 38106 Braunschweig, Germany*
** *Center of Pharmaceutical Engineering, TU Braunschweig, Franz-Liszt-Straße 35a, 38106 Braunschweig, Germany*
*** *Automation & Computer Sciences Department, Harz University of Applied Sciences, Friedrichstr. 57-59, 38855 Wernigerode, Germany (e-mail: rschenkendorf@hs-harz.de)*

Abstract: In (bio)chemical process engineering, first-principles process models have played a central role for some time in better understanding, monitoring, and controlling these complex processes. Dynamic process models have become even more critical in the context of Industry 4.0 and the use of digital twins in the last decade. However, the quality and the technology readiness level of digital process models depend crucially on the reliability of the model predictions. In addition to a suitable model structure/hypothesis, the model parameters of the implemented kinetics are of paramount importance. The accuracy of the parameter estimation, in turn, depends on the quantity and quality of the data as well as on the employed parameter identification solving strategies, where ordinary least squares concepts are still the standard. We propose a novel parameter identification concept that combines systems theory and machine learning principles. The parameter identification problem is formulated as a total least squares optimization problem that uses neural ordinary differential equations for surrogate modeling and recalculates the model control inputs with the algebraic differential flatness framework for model inversion. The usefulness of the proposed concept for more precise kinetic parameters is demonstrated with a simulation study of an enzyme-catalyzed biochemical process, where the total least squares approach leads to lower parameter uncertainties compared to the standard concept based on ordinary least squares using the same amount of data.

Keywords: total least squares, neural ordinary differential equations, differential flatness, model inversion, kinetic parameter estimation, parameter uncertainties

1. INTRODUCTION

The importance of mathematical modeling in process development and refinement entails the demand for efficient and reliable system identification and parameter estimation methods, respectively. In many engineering fields, the time behavior of complex technical systems, including (bio)chemical and pharmaceutical manufacturing processes, can be described by a system of ordinary differential equations (ODEs). The ODEs' parameters, however, are typically unknown and have to be estimated using experimental data (Bhonsale et al., 2020; Villaverde et al., 2021; Wieland et al., 2021).

Within the last decades, intense research has led to a variety of concepts regarding parameter estimation. A standard approach minimizes the sum of squared errors (SSE) between a model prediction and measurement data, where the prediction is calculated by solving the ODE system numerically. The parameters are then adapted until a given minimization criterion is reached (Walter et al., 1997).
Besides collocation methods for solving ODE-based parameter identification problems (Wang et al., 2018), algebraic parameter identification techniques avoid the repeated numerical solution by transforming the ODE problem into a more straightforward algebraic optimization problem. The iteratively refined principal differential analysis (iPDA) approach, for instance, involves the fitting of B-splines to the measured data (Poyton et al., 2006), from which the time derivative information can be obtained and substituted into the model equations. Another example is the differential flatness concept, first introduced by Fliess et al. (1995). In a differential flat system, state and input variables can be expressed as functions of so-called flat outputs and a finite number of their derivatives, also leading to a simplification of the optimization problem. Instead of applying optimal experimental design concepts, i.e., improving the data quantity and quality with new optimized experiments (Abt et al., 2018; Krausch et al., 2019; Nimmegeers et al., 2020), the flatness concepts might result in improved parameter sensitivities and more precise parameter estimates, respectively, without the need for new data (Schenkendorf and Mangold, 2014). This, of course, depends on the quality of the empirical process model. However, other concepts, like e.g. physics-informed neural networks (Raissi et al., 2019), demonstrate the possibility of combining data-driven methods with prior

10.1016/j.ifacol.2022.09.131

process knowledge.
The algebraic parameter identification techniques decisively rely on suitable methods for data fitting, to obtain the required information on the model states and their time derivatives. While there are well investigated procedures, like B-splines, in both iPDA and differential flatness (Poyton et al., 2006; Schenkendorf and Mangold, 2014; Liu et al., 2016), we here explore the application of neural ordinary differential equations, a recently introduced family of deep neural network models (Chen et al., 2018).

Given the challenges of parameter estimation, this study investigates a beneficial cost function definition for the optimization process. We compare the parameter estimation results for an ODE model derived by three different approaches: (i) minimization of the SSE between model prediction and data, referred to as ordinary least squares (OLS), (ii) rearrangement of the model equations to solve for the inputs, where these equations contain the model's flat outputs and their derivatives with respect to time. These terms are derived from a neural ODE that was trained on a given data set, followed by minimization of the SSE between prediction and inputs. This approach is referred to as input least squares (ILS). With (iii) a combination of the two is applied, similar to the concept described by Liu et al. (2016) and referred to as total least squares (TLS).

2. METHODS

This section introduces the different formulations of the parameter identification problem, the differential flatness concept for model inversion, and the notations used for OLS, ILS, and TLS. Then we summarize the basic idea of neural ODEs and how we can use them to back-calculate the control inputs for model inversion in the case of ILS and TLS.

2.1 Parameter identification problem

In this work, the dynamic process models read as:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}), \tag{1a}$$
$$\mathbf{x}(t_0) = \mathbf{x}_0, \tag{1b}$$

where $t \in [t_0, t_0 + t_{\text{end}}]$ is the time, with t_0 as the initial time and t_{end} as the time duration of the simulation, $\mathbf{u} \in \mathbf{R}^{n_u}$ is the vector of the control variables, $\mathbf{p} \in \mathbf{R}^{n_p}$ is the vector of the time-invariant parameters, and $\mathbf{x_d} \in \mathbf{R}^{n_x}$ are the differential states. The initial conditions for the differential states are given by $\mathbf{x}_0$. Eq. (1a) is the model equation with $\mathbf{f} : \mathbf{R}^{n_x \times n_u \times n_p} \to \mathbf{R}^{n_x}$.

Within the OLS approach, the actual parameter identification problem reads as:

$$\hat{\mathbf{p}}^{\text{OLS}} = \arg\min_{\mathbf{p}} \sum_{k=1}^{K} ||\mathbf{y}^{\text{data}}(t_k) - \mathbf{y}(t_k, \mathbf{p})||_2^2, \tag{2}$$

where $||\cdot||_2$ denotes the Euclidean norm, and the model output equation is defined as:

$$\mathbf{y}(t_k, \mathbf{p}) = \mathbf{h}(\mathbf{x}(t_k, \mathbf{p})), \tag{3}$$

with $\mathbf{h} : \mathbf{R}^{n_x} \to \mathbf{R}^{n_y}$, and $\mathbf{y} \in \mathbf{R}^{n_y}$ is the vector of the model output.

Using a model inversion concept, back-calculating model inputs, the ILS-based parameter identification is defined as:

$$\hat{\mathbf{p}}^{\text{ILS}} = \arg\min_{\mathbf{p}} \sum_{k=1}^{K} ||\mathbf{u}^{\text{data}}(t_k) - \mathbf{u}(t_k, \mathbf{p})||_2^2, \tag{4}$$

Evaluating model inputs and outputs alike, the TLS-based parameter identification problem is given as:

$$\hat{\mathbf{p}}^{\text{TLS}} = \arg\min_{\mathbf{p}} \sum_{k=1}^{K} ||\mathbf{z}^{\text{data}}(t_k) - \mathbf{z}(t_k, \mathbf{p})||_2^2, \tag{5}$$

with $\mathbf{z}^{\text{data}}(t_k) = [\mathbf{y}^{\text{data}}(t_k), \mathbf{u}^{\text{data}}(t_k)]$ and $\mathbf{z}(t_k, \mathbf{p}) = [\mathbf{y}(t_k, \mathbf{p}), \mathbf{u}(t_k, \mathbf{p})]$.

In Eqs. (4) and (5), the control inputs, $\mathbf{u}(t_k, \mathbf{p})$, are back-calculated using the differential flatness concept. A process model (Eq. (1)) is termed differentially flat if the following output vector exists:

$$\mathbf{y}^{\text{flat}} = \mathbf{h}^{\text{flat}}(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \ldots, \mathbf{u}^{(s)}, \mathbf{p}), \tag{6}$$

with a finite value $s \in \mathbf{N}$ and the smooth mapping function $\mathbf{y}^{\text{flat}} : \mathbf{R}^{n_x} \times (\mathbf{R}^{n_u})^{s+1} \times \mathbf{R}^{n_p} \to \mathbf{R}^{n_y}$; also referred to as a flat output.

In detail, the reconstructed system states and control inputs read as:

$$\mathbf{x} = \Psi_{\mathbf{x}}(\mathbf{y}^{\text{flat}}, \dot{\mathbf{y}}^{\text{flat}}, \ldots, \mathbf{y}^{\text{flat}^{(r)}}, \mathbf{p}), \tag{7}$$
$$\mathbf{u} = \Psi_{\mathbf{u}}(\mathbf{y}^{\text{flat}}, \dot{\mathbf{y}}^{\text{flat}}, \ldots, \mathbf{y}^{\text{flat}^{(r+1)}}, \mathbf{p}), \tag{8}$$

with the mapping functions $\Psi_{\mathbf{x}} : (\mathbf{R}^{n_y})^{r+1} \times \mathbf{R}^{n_p} \to \mathbf{R}^{n_x}$ and $\Psi_{\mathbf{u}} : (\mathbf{R}^{n_y})^{r+2} \times \mathbf{R}^{n_p} \to \mathbf{R}^{n_u}$, and assuming a quadratic system:

$$\dim \mathbf{y}^{\text{flat}} = \dim \mathbf{u}. \tag{9}$$

When applying the flatness concept, it was shown that parameter sensitivities and the reliability of parameter estimates could be improved in the case of ILS (Schenkendorf and Mangold, 2014) and TLS (Liu et al., 2016), respectively. Note that the flatness concept has also proven beneficial for system identification and model selection problems (Schulze and Schenkendorf, 2020), which is out of the scope of this work.

However, the parameterization of the flat output $\mathbf{y}^{\text{flat}}$ for evaluating Eq. (8) and Eqs. (4), (5), respectively, is still challenging. Standard concepts are based on empirical models using a B-Spline setting with its drawback of calibration and proper use to represent the derivatives of the system states (Eq. (1)) (Poyton et al., 2006; Varziri et al., 2008). Alternatively, so-called neural ordinary differential equations can be used to derive an empirical dynamical system and to define $\mathbf{y}^{\text{flat}}$ and its derivatives. Note that neural ODEs are tailored to represent dynamic processes based on discrete measurement data and straightforward to train due to backpropagation and the augmented adjoint state vector, including time and the neural network's meta-parameters (Chen et al., 2018; Rackauckas et al., 2020; Kim et al., 2021). Recently, neural ODEs are finding their way into the process systems engineering community to solve simulation-based process analysis and control problems (De Jaegher et al., 2021; Francis-Xavier et al., 2021; Arnold and King, 2021).

2.2 Neural ordinary differential equations

In data science, neural networks (e.g., feed-forward neural networks, multilayer perceptron, and recurrent neural networks) are frequently used to build data-driven empirical models. Basically, the i*th* neural network layer, $\mathrm{NNL}_i(x) : \mathbf{R}^{d_{i-1}} \rightarrow \mathbf{R}^{d_i}$, contains N_i neurons. Here, $\mathrm{NNL}_i(x)$ is specified with the weight matrix, $\mathbf{W}^i \in \mathbf{R}^{d_i \times d_{i-1}}$, and the bias vector, $\mathbf{b}_i \in \mathbf{R}^{d_i}$. And thus, for instance, a feed-forward neural network reads as:

$$\mathrm{NNL}_0(x) = x \in \mathbf{R}^{d_0}, \tag{10a}$$

$$\mathrm{NNL}_j(x) = \sigma(\mathbf{W}^j \mathrm{NNL}_{j-1}(x) + \mathbf{b}_j) \in \mathbf{R}^{d_j}; \quad \forall 1 \leq j \leq I-1, \tag{10b}$$

$$\mathrm{NNL}_I(x) = \mathbf{W}^I \mathrm{NNL}_{I-1}(x) + \mathbf{b}_I \in \mathbf{R}^{d_I}, \tag{10c}$$

with the input layer NNL_0, the hidden layer NNL_j, and the output layer NNL_I. Frequently, sigmoid or tangent functions are used for the activation function $\sigma(\cdot)$ (Bishop, 2006).

When it comes to the so-called neural ordinary differential equations, the governing equations read as:

$$\dot{\mathbf{x}}(t) = \mathrm{NN}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}), \tag{11a}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \tag{11b}$$

Note that the neural ODE system (Eq. (10)) can be used to represent data records exclusively or to include process knowledge (i.e., first-principles elements) within a hybrid model framework alike. Moreover, neural ODEs have been successfully used for system identification under sparse and noisy data (Brunton et al., 2016; Vortmeyer-Kley et al., 2021). Furthermore, the neural network's architecture, e.g., the number of hidden layers, activation functions, etc., was not optimized. Such optimization could be added and complemented with optimal experimental design methods for a better representation of the experimental data based on the neural ODEs, which is beyond the scope of this work.

In what follows, all implementations were coded in Julia (Bezanson et al., 2017). For the studied NN and the neural ODEs, the Flux.jl (Innes et al., 2018; Innes, 2018) and the DiffEqFlux.jl (Rackauckas et al., 2019) Julia library were utilized, respectively. The Adam (Kingma and Ba, 2017) first-order gradient-based optimization algorithm was used with the L2-norm (sum of squared errors) to train the NN and the neural ODEs. We ran all simulations on a standard desktop machine.

3. CASE STUDY

The underlying model of this study describes an enzymatic reaction from the substrate benzaldehyde to benzoin under consideration of the intermediate substrate-enzyme complex. To this end, we adapt the process model given in Schulze and Schenkendorf (2020) accordingly. Based on Table 1, the reaction scheme includes benzaldehyde as substrate S_1, benzoin as product P, the intermediate C_1, and the enzyme E. The enzyme reaction chain might be subject to a loss reaction with an irreversible inhibition due to the second substrate S_2 leading to the loss product C_2. The kinetic constants are k_1, k_2, and k_3.

Table 1. Reaction scheme of the process model.

Model $\mathcal{M}$
$2\,\mathrm{S}_1 + \mathrm{E} \xrightarrow{k1} \mathrm{C}_1 \xrightarrow{k2} \mathrm{P} + \mathrm{E}$
$\mathrm{S}_2 + \mathrm{E} \xrightarrow{k3} \mathrm{C}_2$

The governing equations of the related ODE systems read as:

$$\mathcal{M} = \begin{cases} \dot{\mathrm{S}}_1 &= -2k_1 {\mathrm{S}_1}^2 \mathrm{E} + u_1 \\ \dot{\mathrm{C}}_1 &= k_1 {\mathrm{S}_1}^2 \mathrm{E} - k_2 \mathrm{C}_1 + u_2 \\ \dot{\mathrm{E}} &= -k_1 {\mathrm{S}_1}^2 \mathrm{E} + k_2 \mathrm{C}_1 - k_3 [\mathrm{S}_2] \mathrm{E} + u_3 \\ \dot{\mathrm{P}} &= k_2 \mathrm{C}_1 + u_4 \end{cases}, \tag{12}$$

In model $\mathcal{M}$, please note that the differential equations of the second substrate S_2, which is assumed to be constant, as well as the loss product C_2 are not given, as information about their time response would not affect the other differential equations and is consequently not relevant to the problem at hand.

Note that the theoretical control vector dimension is 4, however, only u_1 and u_3 are active. Thus, u_1 and u_4 remain zero for the considered scenario. The dimension of the flat output vector, in turn, must be 4 according to Eq. (9).

We infer the flat outputs by a heuristic method to 1) find a flat output candidate in accordance with Equation (6) and 2), use graph theory to prove that the candidate fulfills the differential flatness conditions given Equations (7)–(9).

To illustrate the information flow and connectivity, the corresponding adjacency matrices with $\mathbf{x} = \{\mathrm{S}_1, \mathrm{C}_1, \mathrm{E}, \mathrm{P}\}$ are:

$$A_u = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{array}{c} \begin{array}{cccc} u_1 & u_2 & u_3 & u_4 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}, \tag{13}$$

$$A_x = \begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \end{array}, \tag{14}$$

$$A_y = \begin{array}{c} \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & x_3 & x_4 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}, \tag{15}$$

and the resulting digraph is shown in Fig. 1. The digraph is composed of 12 vertices, $V = \{u_1, u_2, u_3, u_4\} \cup \{x_1, x_2, x_3, x_4\} \cup \{y_1, y_2, y_3, y_4\}$, and 17 edges corresponding to the non-zero entries in the adjacency matrices A_u, A_x, and A_y. The self-loops of $\{x_1, x_2, x_3\} \in V$ are related to the non-zero diagonal elements of A_x.

The resulting digraph (see Fig. 1) shows the trivial solution for a proper flat output selection

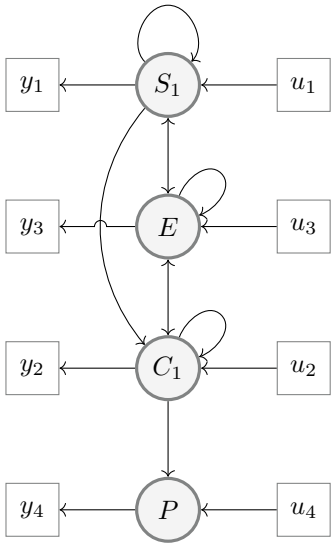

Fig. 1. Digraph for model $\mathcal{M}$ to study the differential flatness property

$$\mathbf{y}^{\text{flat}} = \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{pmatrix} = \begin{pmatrix} \mathrm{S}_1 \\ \mathrm{C}_1 \\ \mathrm{E} \\ \mathrm{P} \end{pmatrix} \tag{16}$$

that satisfies Eq. (6).

From the ODE system, after the system states substitutions, we obtain the inverse model:

$$\mathcal{M}^{-1} = \begin{cases} \mathrm{S}_1 & = \gamma_1 \\ \mathrm{C}_1 & = \gamma_2 \\ \mathrm{E} & = \gamma_3 \\ \mathrm{P} & = \gamma_4 \\ u_1 & = \dot{\gamma}_1 + 2k_1\gamma_1^2\gamma_3 \\ u_2 & = \dot{\gamma}_2 + k_2\gamma_2 - k_1\gamma_1^2\gamma_3 \\ u_3 & = \dot{\gamma}_3 + k_1\gamma^2\gamma_3 - k2\gamma_2 + k_3\mathrm{S}_2\gamma_3 \\ u_4 & = \dot{\gamma}_4 - k_2\gamma_2 \end{cases} \tag{17}$$

To derive the flat outputs and its derivatives we implement the neural ODE system according to:

$$\dot{\gamma} = NN(\gamma_1, \gamma_2, \gamma_3, \gamma_4) \tag{18}$$

with $\dot{\gamma} = [\dot{\gamma_1}, \dot{\gamma_2}, \dot{\gamma_3}, \dot{\gamma_4}]^T$, and where $NN(\cdot)$ refers to a multilayer perceptron (MLP) with two hidden layers, i.e., 32 and 16 nodes per hidden layer, respectively. Input and output layer with 4 nodes each (Eq. (10)), and *Swish* as activation function (Ramachandran et al., 2017). For the sake of simplicity, we use noise-free *in silico* data obtained by a simulation of the model with predefined kinetic parameters and a sampling rate of 0.1 s over a time span of 5 s. All states were considered measurable in units of $\mathrm{mmol\,L^{-1}}$. The data was calculated with parameter values of 0.2 $\mathrm{L^2\,mmol^{-2}\,min^{-1}}$, 2.0 $\mathrm{min^{-1}}$ and 0.02 $\mathrm{L\,mmol^{-1}\,min^{-1}}$ for k_1, k_2 and k_3, respectively. This data was also used to train the neural ODEs.

For the parameter estimation $n = 1000$ starting vectors $\hat{k}$ were generated by random realizations (i.e., Gaussian distribution) of the parameters k_1, k_2, k_3. Each vector was then used to perform a parameter estimation based on the three different cost functions for parameter identification (i.e., OLS, ILS, and TLS) given in Eqs. (2) to (5), respectively. Please note that the neural network does not have to be adapted for the iterative parameter estimation procedure. The required information on the model states and their time derivatives for ILS and TLS is obtained from a single training data set. Within the evaluation of our study we here focus on the distribution of the parameter estimates regarding each approach. Therefore, z-scores of the resulting parameter vectors were calculated to allow a direct comparison, i.e., zero-mean parameter estimates normalized to the standard variance related to OLS are used.

4. RESULTS

In Fig. 2, we show z-score transformed parameter estimates for k_1, k_2, and k_3. For all parameter combinations, the proposed TLS approach reveals the lowest rate of scattering. The exclusive use of back-calculated inputs within the ILS concept seems to be beneficial for k_1 and k_2 compared to OLS; see Fig. 2a. However, in the case of kinetic parameter k_3, ILS leads to the highest scattering; see Figs. 2b and 2c. Please also note that for all concepts (i.e., OLS, ILS, and TLS) parameter correlations are low.

Moreover, when analyzing the individual parameter uncertainties using kernel density estimators as in Fig. 3, first, we see Gaussian-type probability density functions for all kinetic parameters independently of the used parameters identification concept. Based on this, we can conclude that the underlying parameter identification problems are well-posed for OLS, ILS, and TLS, and the kinetic parameters are identifiable - at least locally. While the estimated parameter values k_1 and k_2 are distributed quite similarly in ILS and TLS (Figs. 3a, 3b), the ILS approach led to the highest level of variation regarding the estimates for parameter k_3; see Fig. 3c. Obviously, OLS and ILS might be beneficial for certain subsets of the kinetic parameters; TLS combines the positive individual properties of OLS and ILS, and thus, leads to the most accurate parameter estimates globally. Please also note that ILS and TLS depend critically on the quality of the back-calculated flat outputs and their derivatives based on the used neural ODE specifications. However, a systematic study analyzing the effect of the neural ODE's meta-parameters is out of the scope of this contribution.

5. CONCLUSIONS

The usefulness of digital twins and model-based concepts in process systems engineering depends essentially on the reliability and precision of the estimated kinetic model parameters. In this work, we successfully demonstrated that a parameter identification problem, which evaluates total least squares (TLS) instead of ordinary least squares (OLS), ensures more precise parameter estimates in terms of parameter scattering. Here, our original contribution is the proper combination of advanced systems theory concepts (i.e., differentially flatness) and recent developments in data science with neural ordinary differential equations. As a case study, a biotechnology process model

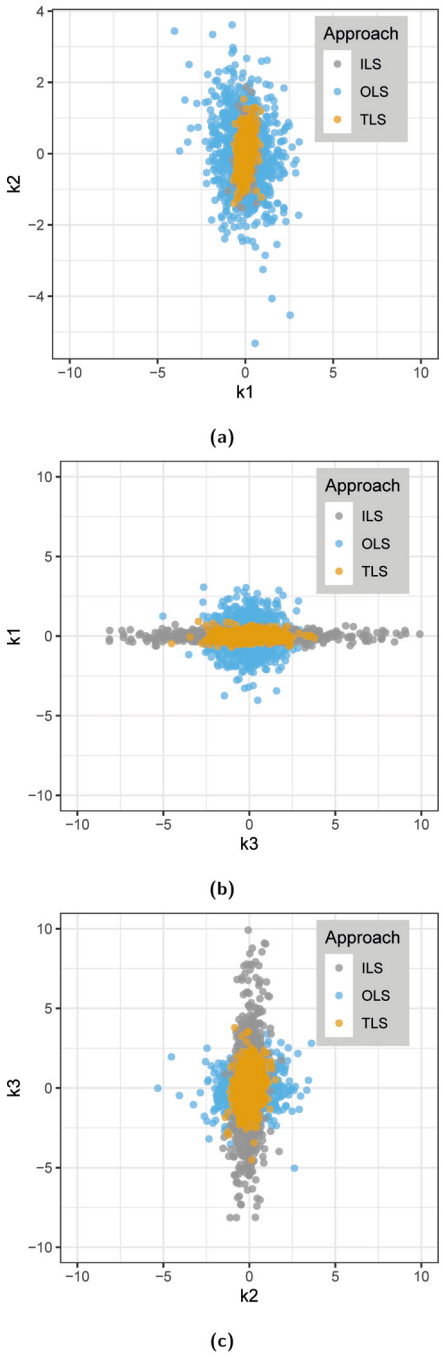

Fig. 2. Scatter plots of z-sore transformed parameter estimates for OLS, ILS and TLS obtained from $n = 1000$ optimizations.

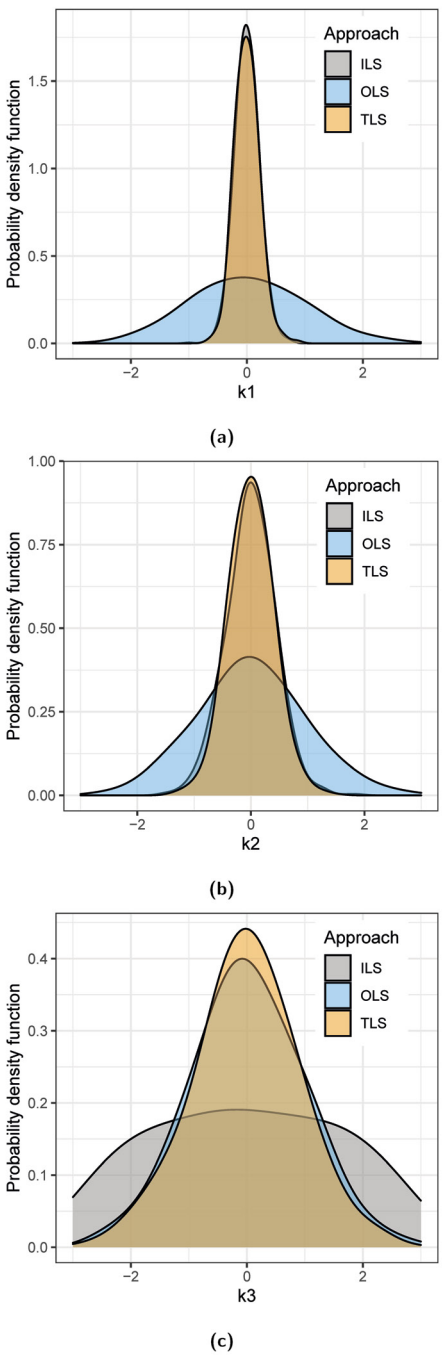

Fig. 3. Marginal probability density plot of z-score transformed parameter estimates for OLS, ILS and TLS obtained from $n = 1000$ optimizations.

was implemented, and related kinetic parameters were identified. Moreover, the neural ODE concept ensured an efficient parameterization of the flat output and its derivatives - though, the neuronal ODEs have not been systematically studied in terms of training and calibration. The integrated consideration of first-principles parameters and the meta-parameters of neuronal ODEs will be part of future work, including the effect of sparse and noisy experimental data.

ACKNOWLEDGEMENTS

We gratefully acknowledge the support by the German Research Foundation (444703025). M.S. is also grateful to the International Max Planck Research School “Advanced Methods in Process and Systems Engineering” (IMPRS ProEng).

REFERENCES

Abt, V., Barz, T., Cruz-Bournazou, M.N., Herwig, C., Kroll, P., Möller, J., Pörtner, R., and Schenkendorf, R. (2018). Model-based tools for optimal exper-

iments in bioprocess engineering. *Current Opinion in Chemical Engineering*, 22, 244–252. doi: 10.1016/j.coche.2018.11.007.

Arnold, F. and King, R. (2021). State–space modeling for control based on physics-informed neural networks. *Engineering Applications of Artificial Intelligence*, 101(July 2020), 104195. doi: 10.1016/j.engappai.2021.104195.

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V.B. (2017). Julia: A fresh approach to numerical computing. *SIAM review*, 59(1), 65–98.

Bhonsale, S.S., Stokbroekx, B., and Van Impe, J. (2020). Assessment of the parameter identifiability of population balance models for air jet mills. *Computers & Chemical Engineering*, 143, 107056.

Bishop, C.M. (2006). *Pattern recognition and machine learning*. Springer-Verlag New York.

Brunton, S.L., Proctor, J.L., and Kutz, J.N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15), 3932–3937. doi:10.1073/pnas.1517384113.

Chen, R.T.Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. (2018). Neural Ordinary Differential Equations. (NeurIPS). doi:arXiv:1806.07366v3. URL `arxiv.org/abs/1806.07366`.

De Jaegher, B., De Schepper, W., Verliefde, A., and Nopens, I. (2021). Enhancing mechanistic models with neural differential equations to predict electrodialysis fouling. *Separation and Purification Technology*, 259(July 2020), 118028. doi: 10.1016/j.seppur.2020.118028.

Fliess, M., Lévine, J., Martin, P., and Rouchon, P. (1995). Flatness and defect of non-linear systems: introductory theory and examples. *International Journal of Control*, 61, 13–27. doi:10.1080/00207179508921959.

Francis-Xavier, F., Kubannek, F., and Schenkendorf, R. (2021). Hybrid process models in electrochemical syntheses under deep uncertainty. *Processes*, 9(4), 704. doi: 10.3390/pr9040704.

Innes, M., Saba, E., Fischer, K., Gandhi, D., Rudilosso, M.C., Joy, N.M., Karmali, T., Pal, A., and Shah, V. (2018). Fashionable modelling with flux. *CoRR*, abs/1811.01457. URL `arxiv.org/abs/1811.01457`.

Innes, M. (2018). Flux: Elegant machine learning with julia. *Journal of Open Source Software*. doi: 10.21105/joss.00602.

Kim, S., Ji, W., Deng, S., Ma, Y., and Rackauckas, C. (2021). Stiff neural ordinary differential equations. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(9), 093122. doi:10.1063/5.0060697.

Kingma, D.P. and Ba, J. (2017). Adam: A method for stochastic optimization.

Krausch, N., Barz, T., Sawatzki, A., Gruber, M., Kamel, S., Neubauer, P., and Cruz Bournazou, M.N. (2019). Monte Carlo Simulations for the Analysis of Non-linear Parameter Confidence Intervals in Optimal Experimental Design. *Frontiers in Bioengineering and Biotechnology*, 7(May), 1–16. doi:10.3389/fbioe.2019.00122.

Liu, J., Mendoza, S., Li, G., and Fathy, H. (2016). Efficient total least squares state and parameter estimation for differentially flat systems. In *2016 American Control Conference (ACC)*, 5419–5424. doi: 10.1109/ACC.2016.7526519.

Nimmegeers, P., Bhonsale, S., Telen, D., and Van Impe, J. (2020). Optimal experiment design under parametric uncertainty: A comparison of a sensitivities based approach versus a polynomial chaos based stochastic approach. *Chemical Engineering Science*, 221, 115651. doi:10.1016/j.ces.2020.115651.

Poyton, A., Varziri, M., McAuley, K., Mclellan, P., and Ramsay, J. (2006). Parameter estimation in continuous-time dynamic models using principal differential analysis. *Computers & Chemical Engineering*, 30, 698–708. doi:10.1016/j.compchemeng.2005.11.008.

Rackauckas, C., Innes, M., Ma, Y., Bettencourt, J., White, L., and Dixit, V. (2019). Diffeqflux.jl-a julia library for neural differential equations. *arXiv preprint arXiv:1902.02376*.

Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., and Edelman, A. (2020). Universal differential equations for scientific machine learning.

Raissi, M., Perdikaris, P., and Karniadakis, G. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686–707. doi: doi.org/10.1016/j.jcp.2018.10.045.

Ramachandran, P., Zoph, B., and Le, Q.V. (2017). Searching for activation functions. *CoRR*, abs/1710.05941. URL `arxiv.org/abs/1710.05941`.

Schenkendorf, R. and Mangold, M. (2014). Parameter identification for ordinary and delay differential equations by using flat inputs. *Theoretical Foundations of Chemical Engineering*, 48(5), 594–607.

Schulze, M. and Schenkendorf, R. (2020). Robust model selection: Flatness-based optimal experimental design for a biocatalytic reaction. *Processes*, 8(2). doi: 10.3390/pr8020190.

Varziri, M., Poyton, A., McAuley, K., Mclellan, P., and Ramsay, J. (2008). Selecting optimal weighting factors in ipda for parameter estimation in continuous-time dynamic models. *Computers & Chemical Engineering*, 32, 3011–3022. doi:10.1016/j.compchemeng.2008.04.005.

Villaverde, A.F., Pathirana, D., Fröhlich, F., Hasenauer, J., and Banga, J.R. (2021). A protocol for dynamic model calibration.

Vortmeyer-Kley, R., Nieters, P., and Pipa, G. (2021). A trajectory-based loss function to learn missing terms in bifurcating dynamical systems. *Scientific Reports*, 20394. doi:10.1038/s41598-021-99609-x.

Walter, E., Norton, J., and Pronzato, L. (1997). *Identification of Parametric Models: From Experimental Data*. Communications and Control Engineering. Springer.

Wang, Y., Biegler, L.T., Patel, M., and Wassick, J. (2018). Parameters estimation and model discrimination for solid-liquid reactions in batch processes. *Chemical Engineering Science*, 187, 455–469. doi: 10.1016/j.ces.2018.05.040.

Wieland, F.G., Hauber, A.L., Rosenblatt, M., Tönsing, C., and Timmer, J. (2021). On structural and practical identifiability. *Current Opinion in Systems Biology*.