Parameter Identification Concept for Process Models Combining Systems Theory and Deep Learning [†]

Subiksha Selvarajan [1], Aike Aline Tappe [1], Caroline Heiduk [2], Stephan Scholl [2] and René Schenkendorf [1,*]

[1] Automation & Computer Sciences Department, Harz University of Applied Sciences, Friedrichstr. 57-59, 38855 Wernigerode, Germany; s.selvarajan@tu-braunschweig.de (S.S.); aike.a.tappe@campus.tu-berlin.de (A.A.T.)

[2] Institute for Chemical and Thermal Process Engineering, TU Braunschweig, Langer Kamp 7, 38106 Braunschweig, Germany; c.heiduk@tu-braunschweig.de (C.H.); s.scholl@tu-braunschweig.de (S.S.)

[*] Correspondence: rschenkendorf@hs-harz.de

[†] Presented at the 1st International Electronic Conference on Processes: Processes System Innovation, 17–31 May 2022; Available online: https://ecp2022.sciforum.net.

Abstract: In recent years, dynamic process models have grown even more important in the context of Industry 4.0 and the use of digital twins. However, the accuracy of corresponding model parameter estimates is determined by the quantity and quality of data and the parameter identification solving methodologies used. Standard methods are based on the ordinary least squares framework. Still, other options are available that might be more sensitive to model parameter variations and ensure more precise parameter estimates. The paper presents a novel technique for parameter identification based on incorporating neural ordinary differential equations for surrogate modeling and differential flatness, i.e., a systems theory concept in control engineering. This approach may lead to improved parameter sensitivities, as demonstrated with a simulation study of a distributer-parameter identification problem assuming a diffusion-type parabolic partial differential equation.

Keywords: Process Systems Engineering; System Identification; Systems Theory; Differential Flatness; Deep Learning; Partial Differential Equations; Parameter Sensitivities

1. Introduction

System identification is the process of creating a mathematical model, or equation, to represent a real-world problem. This equation can then be used to predict and analyze possible outcomes of the system under study. In many engineering fields, the time behavior of complicated technical systems can be described by using a system of ordinary differential equations (ODEs). However, the parameters in these equations are often unknown and need to be estimated from experimental data. Over the past few decades, there has been much research into parameter estimation methods. A popular method minimizes the sum of squared errors (SSE) between a model prediction and measurement data, where the prediction is calculated by solving the ODE numerically [1]. The model parameters are then adjusted until a given minimization criterion is reached. However, other options are available that might be more sensitive to model parameter variations and might ensure more precise parameter estimates, respectively. Control and systems theory can improve parameter identification procedures, e.g., online parameter identification concepts [2,3].

Another example is differential flatness to recalculate control trajectory profiles for desired system dynamics [4,5], i.e., following a system inversion concept. In a differential flat system, state variables 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 parameter identification problem. Moreover, while optimal experimental design concepts

might be needed to improve data quantity and quality [3], the flatness concepts involved in the flat output approach may result in improved parameter sensitivities [6] and more precise parameter estimates [7,8] without experimental data enrichment.

2. Methods

2.1. Parameter identification problem

Frequently, dynamic process models are given as ordinary differential equation systems:

$$
\begin{aligned}
\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}), \\
\mathbf{x}(t_0) &= \mathbf{x}_0,
\end{aligned}
\tag{1}
$$

where $t \in [\mathbf{t}_0, \mathbf{t}_0 + \mathbf{t}_{\text{end}}]$ is the time, with $\mathbf{t}_0$ as the initial time and $\mathbf{t}_{\text{end}}$ as the time duration of the simulation, $\mathbf{u} \in \mathbf{R}^{\mathbf{n}_{\mathbf{u}}}$ is the vector of the control variables, $\mathbf{p} \in \mathbf{R}^{\mathbf{n}_{\mathbf{p}}}$ is the vector of the time-invariant parameters, and $\mathbf{x} \in \mathbf{R}^{\mathbf{n}_{\mathbf{x}}}$ are the differential system states. The initial conditions for the differential states are given by $\mathbf{x}_0$. Moreover, $\mathbf{f} \colon \mathbf{R}^{\mathbf{n}_{\mathbf{x}} \times \mathbf{n}_{\mathbf{u}} \times \mathbf{n}_{\mathbf{p}}} \to \mathbf{R}^{\mathbf{n}_{\mathbf{x}}}$ represents the corresponding vector field. For this kind of mathematical representation, the standard approach of parameter identification, i.e., the ordinary least squares (OLS) method, can be defined as:

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

where $|| \cdot ||_2$ denotes the Euclidean norm, $\mathbf{y}^{\text{data}}(t_k)$ represents the data vector at discrete time points t_k over all measurement samples K, and the model output function is defined as:

$$
\mathbf{y}(\mathbf{t}_{\mathbf{k}}, \mathbf{p}) = \mathbf{h}(\mathbf{x}(\mathbf{t}_{\mathbf{k}}, \mathbf{p})),
\tag{3}
$$

with $\mathbf{h} \colon \mathbf{R}^{\mathbf{n}_{\mathbf{x}}} \to \mathbf{R}^{\mathbf{n}_{\mathbf{y}}}$, and $\mathbf{y} \in \mathbf{R}^{\mathbf{n}_{\mathbf{y}}}$ as the model output vector. Alternatively, when aiming to utilize the inverse model response, i.e., applying a model inversion strategy, an input least squares (ILS) based parameter identification problem can be used:

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

Here, the control inputs, $\mathbf{u}(\mathbf{t}_{\mathbf{k}}, \mathbf{p})$, have to be calculated to solve the parameter identification problem, and $\mathbf{u}^{\text{data}}(t_k)$ represents the recorded physical input actions. For this purpose, in this work, we study the differential flatness concept as outlined in Section 2.2. However, it is essential to note that parameter sensitivities are relevant for well-posed parameter identification problems. On having the output function $\mathbf{y}(t_k, \mathbf{p})$ and the inputs $\mathbf{u}(t_k, \mathbf{p})$ (inverse model response), the sensitivity S of the parameter p is defined as:

$$
S_{y_{\mathbf{p}}}(t_k) = \frac{\partial \mathbf{y}(t_k, \mathbf{p})}{\partial \mathbf{p}}
\tag{5}
$$

$$
S_{u_{\mathbf{p}}}(t_k) = \frac{\partial \mathbf{u}(t_k, \mathbf{p})}{\partial \mathbf{p}}
\tag{6}
$$

And here, in general, absolute high parameter sensitivity values ensure precise parameter estimates according to the Fisher Information matrix and the Cramér Rao inequality [1,3].

2.2. Differential flatness

In literature, a process model (Eq. (1)) is called differentially flat if there exists an output function:

$$
\mathbf{y}^{\text{flat}} = \mathbf{h}^{\text{flat}}\left(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \dots, \mathbf{u}^{(s)}, \mathbf{p}\right),
\tag{7}
$$

with a finite value $\mathbf{s} \in \mathbf{N}$ and the smooth mapping function $\mathbf{y}^{\text{flat}}: \mathbf{R}^{\mathbf{n_x}} \times (\mathbf{R}^{\mathbf{n_u}})^{\mathbf{s+1}} \times \mathbf{R}^{\mathbf{n_p}} \to \mathbf{R}^{\mathbf{n_y}}$ that is called flat output in the literature. With the flat output, the system states and control inputs are expressed as:

$$\mathbf{x} = \mathbf{\Psi_x}\left(\mathbf{y}^{\text{flat}}, \dot{\mathbf{y}}^{\text{flat}}, \dots, \mathbf{y}^{\text{flat}\,(\mathbf{r})}, \mathbf{p}\right), \tag{8}$$

$$\mathbf{u} = \mathbf{\Psi_u}\left(\mathbf{y}^{\text{flat}}, \dot{\mathbf{y}}^{\text{flat}}, \dots, \mathbf{y}^{\text{flat}\,(\mathbf{r+1})}, \mathbf{p}\right), \tag{9}$$

with the mapping functions $\mathbf{\Psi_x}: (\mathbf{R}^{\mathbf{n_y}})^{\mathbf{r+1}} \times \mathbf{R}^{\mathbf{n_p}} \to \mathbf{R}^{\mathbf{n_x}}$ and $\mathbf{\Psi_u}: (\mathbf{R}^{\mathbf{n_y}})^{\mathbf{r+2}} \times \mathbf{R}^{\mathbf{n_p}} \to \mathbf{R}^{\mathbf{n_u}}$, and assuming a quadratic system $\dim \mathbf{y}^{\text{flat}} = \dim \mathbf{u}$. When applying the flatness concept, it was shown that parameter sensitivities and the reliability of parameter estimates, respectively, could be improved in the case of ILS [6] or when combining the OLS with ILS [7,8]. However, in process systems engineering for instance, besides lumped-parameter systems (i.e., ordinary differential equations) distributed-parameter systems described via partial differential equations are frequently applied. In this case, the differential flatness approach has to be generalized [5,9–11].

2.3. Neural ordinary differential equations

In data science and deep learning, neural networks are frequently used to build empirical models. A neural network is a group of interconnected neurons with one or more hidden layers depending on the network's specific task. Technically, the ith neural network layer, $NNL_i(x): R^{d_{i-1}} \to R^{d_i}$, contains N_i neurons. Here, $NNL_i(x)$ is specified with the weight matrix, $W^i \in R^{d_i \times d_{i-1}}$, and the bias vector, $b_i \in R^{d_i}$. And thus, for instance, a feed-forward neural network reads as:

$$\begin{aligned} NNL_0(x) &= x \in R^{d_0}, \\ NNL_j(x) &= \sigma\left(W^j NNL_{j-1}(x) + b_j\right) \in R^{d_j} \\ &\qquad \forall 1 \le j \le I-1 \\ NNL_I(x) &= W^I NNL_{I-1}(x) + b_I \in R^{d_I}. \end{aligned} \tag{10}$$

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

$$\dot{\mathbf{x}}(\mathbf{t}) = \mathbf{NN}(\mathbf{x}(\mathbf{t}), \mathbf{u}(\mathbf{t}), \mathbf{p}), \ \mathbf{x}(\mathbf{t_0}) = \mathbf{x_0}. \tag{11}$$

Neural ODEs offer a promising approach for hybrid modeling and system identification [12–15]. Furthermore, the neural network's architecture can be optimized to represent experimental data better. This could be done in conjunction with optimal experimental design methods [3] to improve the accuracy of system identification further.

3. Case Study

Determining kinetic parameters of (diffusion-type parabolic) PDEs (e.g., Eq. (12)) has been extensively studied, including chromatography and adsorption processes [2,16–19], respectively.

$$\frac{\partial \phi}{\partial t} = \mathbf{p} \frac{\partial^2 \phi}{\partial x^2} + \mathbf{u}(\mathbf{x}, \mathbf{t}) \tag{12}$$

Following the numerical solution and the finite difference method (Eq. (13)) of the diffusion-type PDE, a set of coupled ODEs is obtained, which can be written in the state-space form as shown in Equation (14).

$$\frac{\partial^2 \phi}{\partial x^2} \simeq \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2} \tag{13}$$

$$\frac{\partial \phi_1}{\partial t} = \frac{p}{\Delta x^2}\phi_2 - \frac{2p}{\Delta x^2}\phi_1 + \frac{p}{\Delta x^2}\phi_0 + u_1(t),$$

$$\frac{\partial \phi_2}{\partial t} = \frac{p}{\Delta x^2}\phi_3 - \frac{2p}{\Delta x^2}\phi_2 + \frac{p}{\Delta x^2}\phi_1 + u_2(t),$$

$$\frac{\partial \phi_3}{\partial t} = \frac{p}{\Delta x^2}\phi_4 - \frac{2p}{\Delta x^2}\phi_3 + \frac{p}{\Delta x^2}\phi_2 + u_3(t), \tag{14}$$

$$\ldots$$

$$\frac{\partial \phi_{N-1}}{\partial t} = \frac{p}{\Delta x^2}\phi_N - \frac{2p}{\Delta x^2}\phi_{N-1} + \frac{p}{\Delta x^2}\phi_{N-2} + u_{N-1}(t),$$

$$\frac{\partial \phi_N}{\partial t} = \frac{p}{\Delta x^2}\phi_{N+1} - \frac{2p}{\Delta x^2}\phi_N + \frac{p}{\Delta x^2}\phi_{N-1} + u_N(t).$$

Practically, the parameter identification problem for this academic case study is to determine the diffusion parameter p in this system to represent the actual physical process being modeled accurately. This can be challenging, as even minor changes in the coefficient value can result in significant changes in solution behavior or vice versa. However, it is often possible to obtain reasonable estimates for the diffusion parameter with careful analysis and experimentation [2,18], including the proposed concept of combining systems theory with differential flatness and deep learning. In particular, when assuming that all states in Eq. (14) are measurable, i.e., $y_i = \phi_i, \forall\, 1 \le i \le N$, and that the output derivatives, i.e., $\dot{y}_i, \forall\, 1 \le i \le N$, exist, then the related equation system (Eq. 15) can be transformed to determine the input variables, $u_i, \forall\, 1 \le i \le N$, accordingly.

$$
\begin{pmatrix} \dot{y}_1(t) \\ \dot{y}_2(t) \\ \ldots \\ \dot{y}_{N-1}(t) \\ \dot{y}_N(t) \end{pmatrix} =
\begin{pmatrix}
-\frac{2p}{\Delta x^2} & \frac{p}{\Delta x^2} & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
\frac{p}{\Delta x^2} & -\frac{2p}{\Delta x^2} & \frac{p}{\Delta x^2} & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & \frac{p}{\Delta x^2} & -\frac{2p}{\Delta x^2} & \frac{p}{\Delta x^2} & \cdots & 0 & 0 & 0 & 0 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
0 & 0 & 0 & 0 & \cdots & 0 & \frac{p}{\Delta x^2} & -\frac{2p}{\Delta x^2} & \frac{p}{\Delta x^2} \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & \frac{p}{\Delta x^2} & -\frac{2p}{\Delta x^2}
\end{pmatrix}
\begin{pmatrix} y_1(t) \\ y_2(t) \\ \ldots \\ y_{N-1}(t) \\ y_N(t) \end{pmatrix} +
$$
$$
+ \begin{pmatrix}
1 & 0 & 0 & \cdots & 0 & 0 \\
0 & 1 & 0 & \cdots & 0 & 0 \\
0 & 0 & 1 & \cdots & 0 & 0 \\
\ldots & \ldots & \ldots & \ldots & & \\
0 & 0 & 0 & \cdots & 1 & 0 \\
0 & 0 & 0 & \cdots & 0 & 1
\end{pmatrix}
\begin{pmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \\ \ldots \\ u_{N-1}(t) \\ u_N(t) \end{pmatrix} \tag{15}
$$

Moreover, when dealing with measurement data instead of the full model system (Eq. (15)), the neural ODE framework (Eq. (11)) is used as a surrogate model to approximate the output functions and their derivatives. In particular, the following multilayer perceptron (MLP) setting is used: two hidden layers with 400 nodes each; input and output layer with 198 nodes each; *tanh* as activation function. To train the resulting neural ODE system, simulated data with $t_{i+1} - t_i = 0.1$, $\Delta x = 0.01$, $t \in [0, 2]$, $x \in [0, 1]$, $\Phi(t_0, x) = \sin(2\pi x)$, $\Phi(t, x = 0) = 0$, $\Phi(t, x = 1) = 0$, and $p = 0.1$ are used in dimensionless form. Figure 1(a) shows the model response without any distributed control action, i.e., $u(t) = 0$. The diffusion effect can be seen very clearly, as the initial differences along the spatial axis at the start time $y(t_0, x)$ decrease over the simulation time. Accordingly, a different diffusion parameter p would lead to a different degradation profile, reflecting a corresponding sensitivity of the model. Note that this parameter sensitivity allows, in principle, a practical identification of the model parameter when applying OLS with experimental data and Equation (2). Alternatively, the differential flatness approach can be used to impose a desired process behavior. To this end, the necessary but parameter-dependent calculated input variables can be used for parameter estimation following the mentioned ILS concept and Equation (4). In Figure 1(b), for example, an output profile can be seen in which there are no changes over time in the course of the simulation. Please note that the corresponding input profile was determined using the flatness concept combined with the neural ODE system and the specified training setting.

(**a**)　　　　　　　　　　　　　　　　　　(**b**)
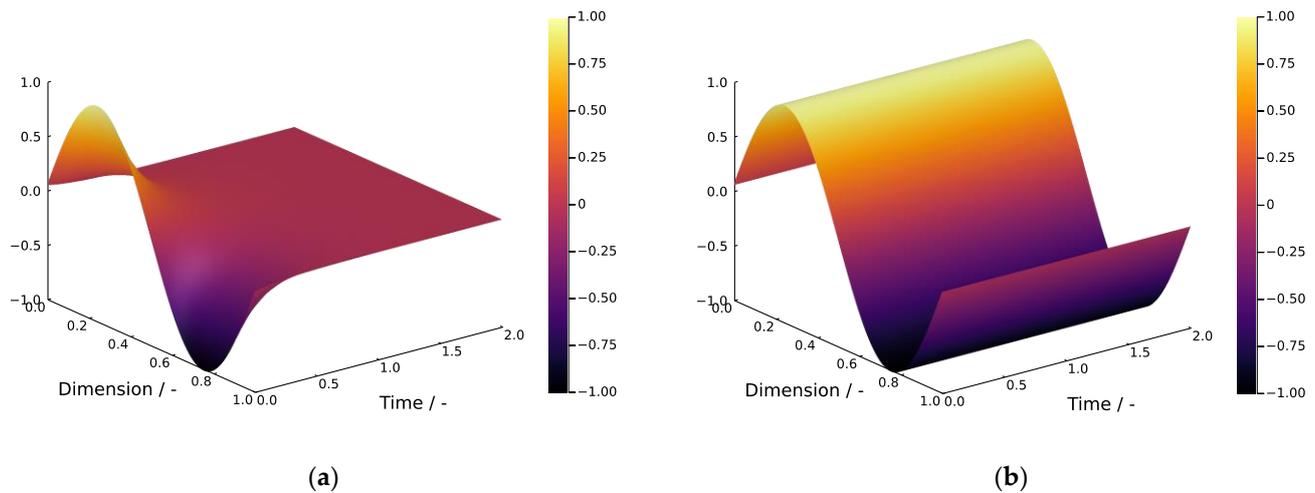
Figure 1. Model response when: (**a**) Zero input profile applied (i.e., zero control action); (**b**) Dedicated input profile applied to compensate for the diffusion effect.

The reconstructed input and the generated output data from Figure 1(b) will be used for the sensitivity analyses. Similar to the output profile, the calculated input profile depends on the diffusion parameter and, thus, is sensitive to its parameter variation. The sensitivities of the output data and the generated control input, corresponding to a variation of the diffusion parameter p, are analyzed using Equations 5 and 6. The resulting sensitivity plots are shown in Figure 2. Here, the output parameter sensitivity (Figure 2(a)) is zero at the starting time, and its absolute values increase at $\mathbf{x} = \mathbf{0}.25$ and $\mathbf{x} = \mathbf{0}.75$, respectively. In the case of the input parameter sensitivity (see Figure 2(b)), these sensitivities are at their peak from the very starting time, and their absolute values are 3-4 times higher than the output parameter sensitivities. Moreover, as mentioned in Section 2, higher parameter sensitivities, in turn, imply better parameter estimates. Consequently, it could be comfortably said that the parameter estimation could be better done using the control input generated by combining the flatness property with the neural ODE concept.

(**a**)　　　　　　　　　　　　　　　　　　(**b**)
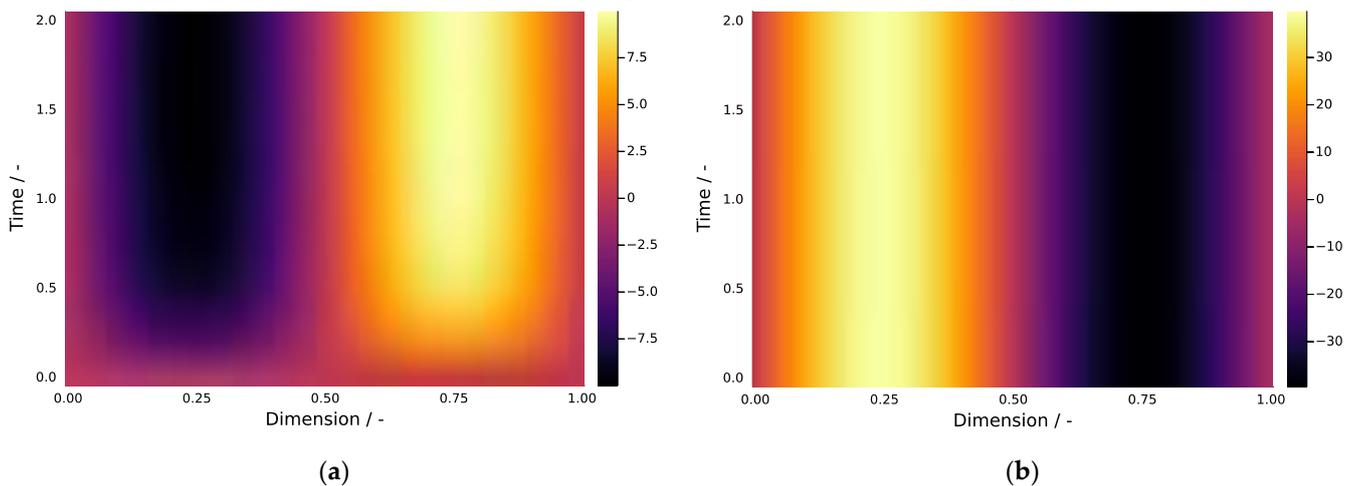
Figure 2. Parameter sensitivity of the diffusion parameter p: (**a**) Using OLS to define the parameter identification problem; (**b**) Using ILS to define the parameter identification problem.

4. Conclusions

Parameter identification is a fundamental problem in systems and control theory. This work successfully demonstrated that a parameter identification problem, which evaluates input least squares (ILS) instead of ordinary least squares (OLS), results in different parameter sensitivities and, in this particular case, an improved parameter sensitivity

range. Here, our original contribution is the proper combination of advanced systems theory concepts (i.e., differential flatness) and recent developments in data science with neural ordinary differential equations. We applied our method to synthetic data generated. Here, we showed that ILS and related parameter sensitivities lead to a significantly higher parameter sensitivity range of the diffusion parameter than OLS-related parameter sensitivity. The improved parameter sensitivity range suggests that ILS may result in better parameter estimates than OLS but might critically depend on the neural ODE system setting and data quality – aspects addressed in ongoing research. Future work will also focus on advanced model inversion schemes which are not limited to differential flat systems.

Author Contributions: Conceptualization R. Schenkendorf.; methodology, S. Selvarajan; implementation, S. Selvarajan, A. Tappe, and R. Schenkendorf; validation, R. Schenkendorf; writing—original draft preparation, A. Tappe.; writing—review and editing, all authors.; visualization, S. Selvarajan; supervision, R. Schenkendorf; funding acquisition, R. Schenkendorf and S. Scholl. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. E. (Eric) Walter, L. Pronzato, Identification of parametric models from experimental data, Springer, New York, 1997.
2. T. Barz, D.C. López C., M.N. Cruz Bournazou, S. Körkel, S.F. Walter, Real-time adaptive input design for the determination of competitive adsorption isotherms in liquid chromatography, Comput. Chem. Eng. 94 (2016) 104–116. doi:10.1016/j.compchemeng.2016.07.009.
3. V. Abt, T. Barz, M.N. Cruz-Bournazou, C. Herwig, P. Kroll, J. Möller, R. Pörtner, R. Schenkendorf, Model-based tools for optimal experiments in bioprocess engineering, Curr. Opin. Chem. Eng. 22 (2018) 244–252. doi:10.1016/j.coche.2018.11.007.
4. M. Fliess, J. Levine, P. Martin, P. Rouchon, Flatness and defect of non-linear systems: introductory theory and examples, Int. J. Control. 61 (1995) 1327–1361. doi:10.1080/00207179508921959.
5. G.G. Rigatos, Nonlinear Control and Filtering Using Differential Flatness Approaches, Springer International Publishing, Cham, 2015. doi:10.1007/978-3-319-16420-5.
6. R. Schenkendorf, M. Mangold, Parameter identification for ordinary and delay differential equations by using flat inputs, Theor. Found. Chem. Eng. 48 (2014) 594–607. doi:10.1134/S0040579514050224.
7. J. Liu, S. Mendoza, G. Li, H. Fathy, Efficient total least squares state and parameter estimation for differentially flat systems, Proc. Am. Control Conf. 2016-July (2016) 5419–5424. doi:10.1109/ACC.2016.7526519.
8. J. Liu, G. Li, H.K. Fathy, A Computationally Efficient Approach for Optimizing Lithium-Ion Battery Charging, J. Dyn. Syst. Meas. Control. 138 (2015) 021009. doi:10.1115/1.4032066.
9. T. Meurer, Flatness-based trajectory planning for diffusion reaction systems in a parallelepiped on spectral approach, Automatica. 47 (2011) 935–949. doi:10.1016/j.automatica.2011.02.004.
10. A. Kater, T. Meurer, Motion planning and tracking control for coupled flexible beam structures, Control Eng. Pract. 84 (2019) 389–398. doi:10.1016/j.conengprac.2018.12.012.
11. T. Meurer, Control of Higher–Dimensional PDEs, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. doi:10.1007/978-3-642-30015-8.
12. K. Lee, E.J. Parish, Parameterized neural ordinary differential equations: Applications to computational physics problems, Proc. R. Soc. A Math. Phys. Eng. Sci. 477 (2021) 1–19. doi:10.1098/rspa.2021.0162.
13. C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, A. Ramadhan, A. Edelman, Universal Differential Equations for Scientific Machine Learning, ArXiv Prepr. (2020). http://arxiv.org/abs/2001.04385.
14. S. Massaroli, M. Poli, J. Park, A. Yamashita, H. Asama, Dissecting Neural ODEs, in: Adv. Neural Inf. Process. Syst. 33, 2020: pp. 3952–3963.
15. N. Sharma, Y.A. Liu, A hybrid science-guided machine learning approach for modeling chemical processes: A review, AIChE J. (2022). doi:10.1002/aic.17609.
16. G. Wang, T. Briskot, T. Hahn, P. Baumann, J. Hubbuch, Estimation of adsorption isotherm and mass transfer parameters in protein chromatography using artificial neural networks, J. Chromatogr. A. 1487 (2017) 211–217. doi:10.1016/j.chroma.2017.01.068.
17. C.S. KUBRUSLY, Distributed parameter system identification A survey†, Int. J. Control. 26 (1977) 509–535. doi:10.1080/00207177708922326.
18. N. Gehring, J. Rudolph, An algebraic algorithm for parameter identification in a class of systems described by linear partial differential equations, PAMM. 16 (2016) 39–42. doi:10.1002/pamm.201610011.
19. J. Grimard, L. Dewasme, A. Vande Wouwer, A Review of Dynamic Models of Hot-Melt Extrusion, Processes. 4 (2016) 19. doi:10.3390/pr4020019.