▲ Hochschule Harz
Harz University of Applied Sciences

# Parameter Identification for Process Models Based on a Combination of Systems Theory and Deep Learning†

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

[1] Automation & Computer Sciences Department, Harz University of Applied Sciences, Friedrichstr. 57-59, 38855 Wernigerode, Germany

[2] Institute for Chemical and Thermal Process Engineering, TU Braunschweig, Langer Kamp 7, 38106 Braunschweig, Germany

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

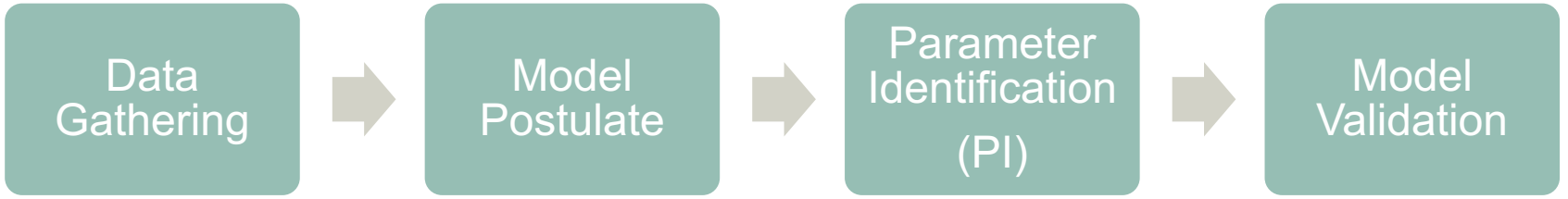
[†] Presented at the 1st International Electronic Conference on Processes: Processes System Innovation, ONLINE, 17-31 May 2022.

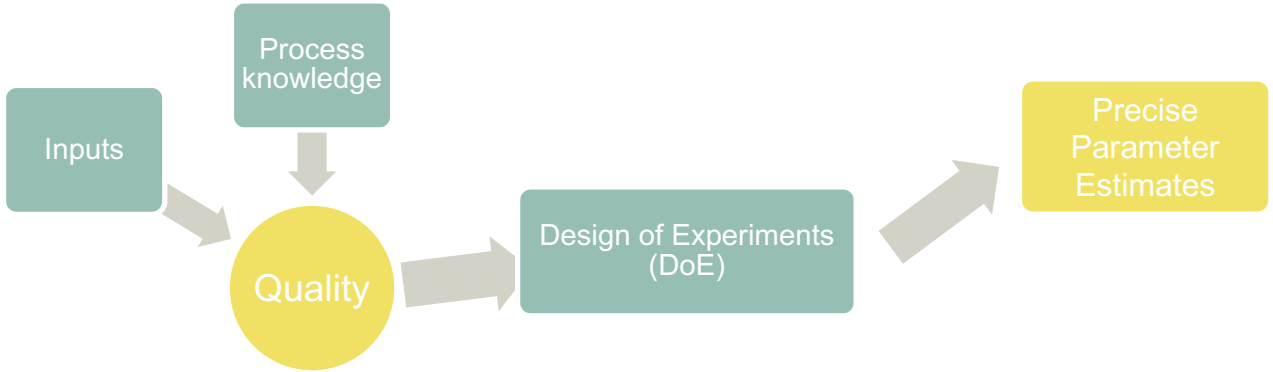
# Outline

- Motivation
- Standard Parameter Identification Framework
- Input-based Parameter Identification Framework
- Input-based Parameter Identification Implementation
- Differential Flatness for Distributed-Parameter Problems
- Case Study: Diffusion-type Problem
- Control Input by Model Inversion
- Parameter Sensitivity Analyses
- Conclusions

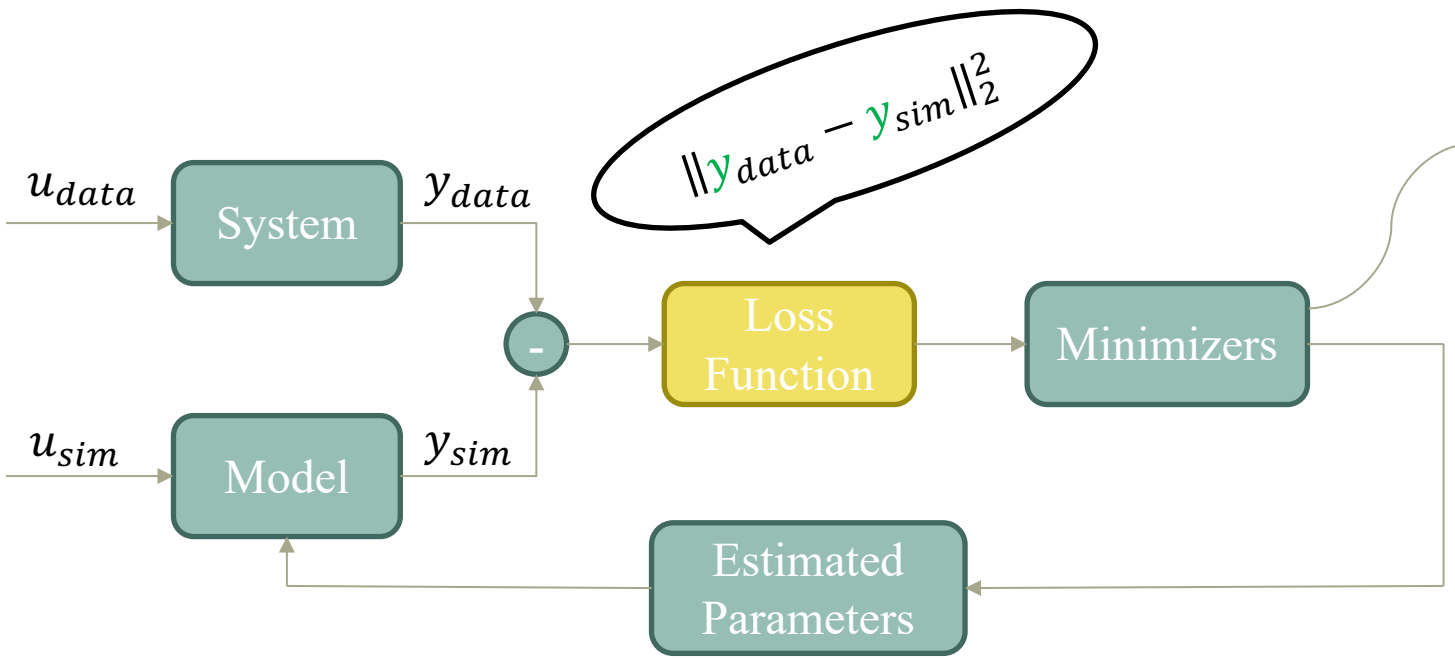
# Motivation

System Identification – can be applied to any industrial process where the inputs and outputs can be measured to build its mathematical model.



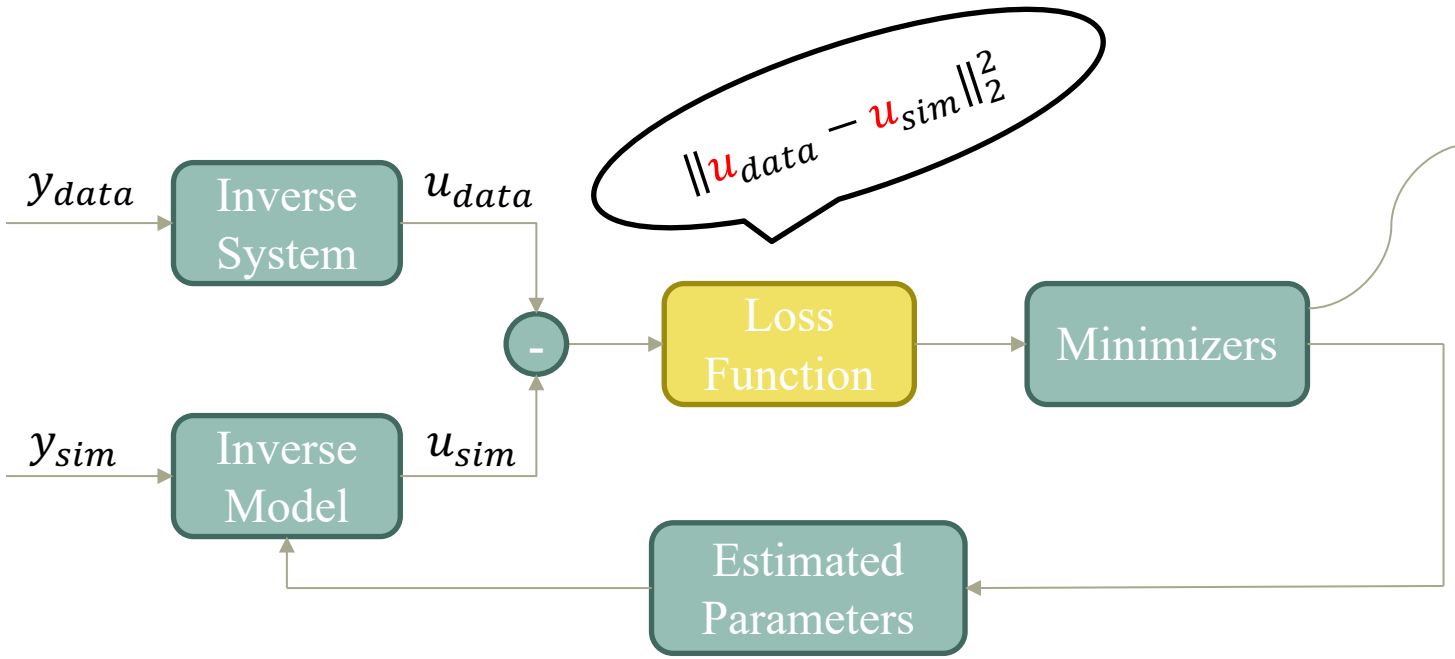



# Standard Parameter Identification Framework




Using local/global optimizers: Simplex, Gradient-and stochastic gradient-based methods, simulated annealing,…

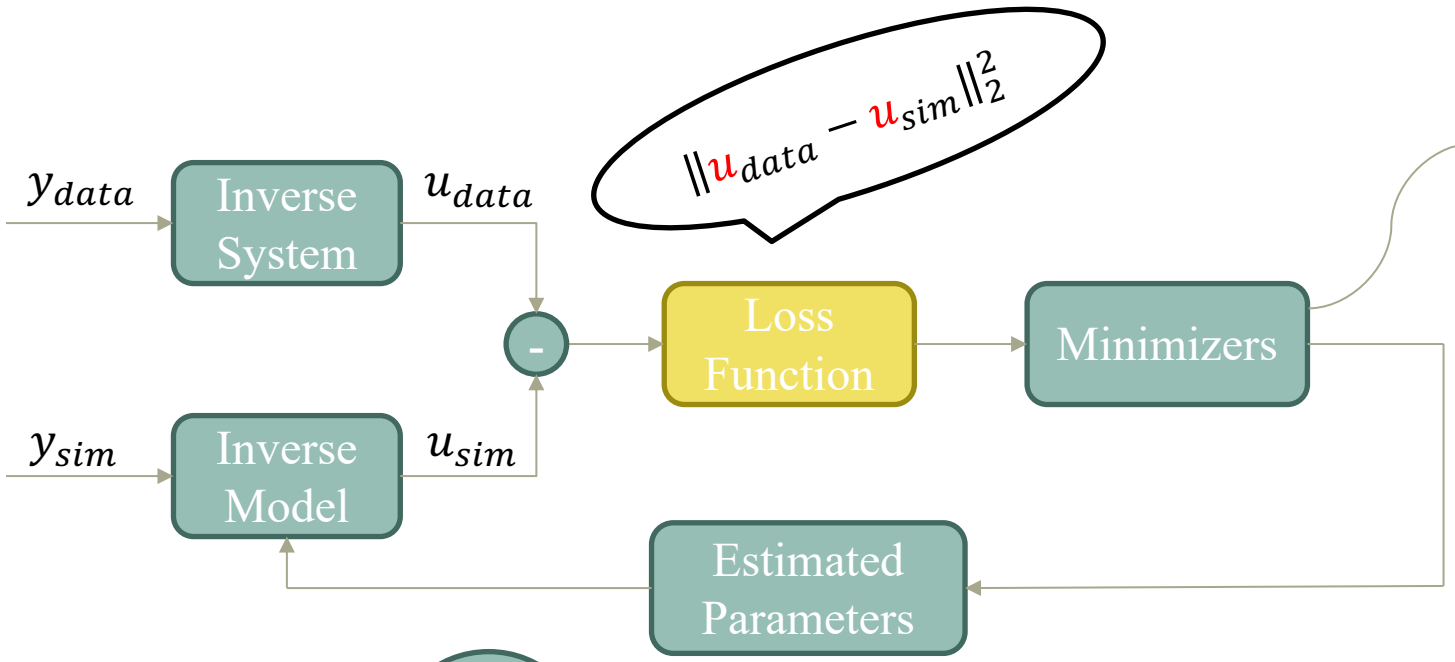
# Input-based Parameter Identification Framework




Using local/global optimizers: Simplex, Gradient-and stochastic gradient-based methods, simulated annealing,…

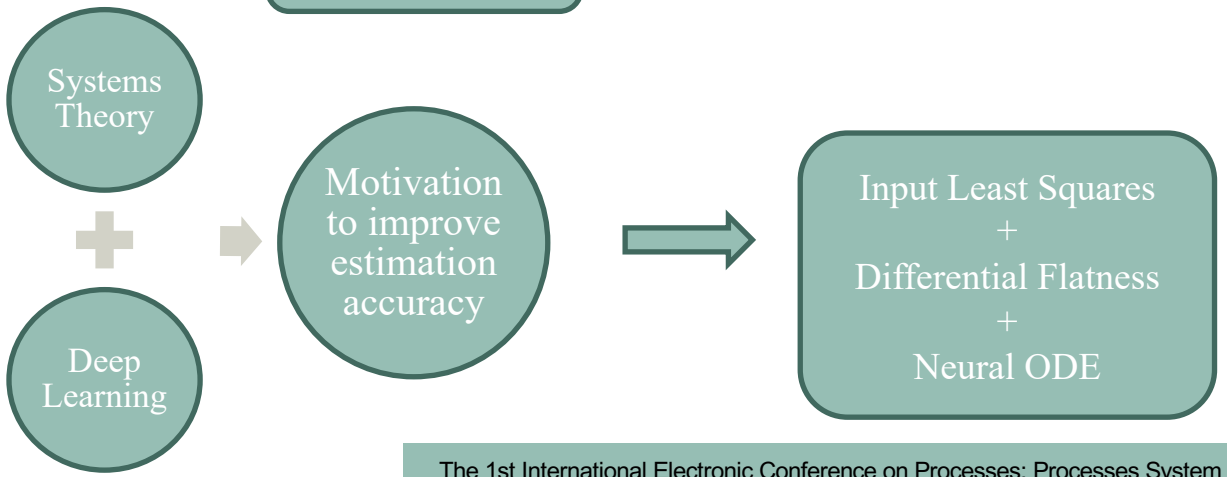
How to get ”inverse model response”?

# Input-based Parameter Identification Implementation Aspects




Using local/global optimizers: Simplex, Gradient-and stochastic gradient-based methods, simulated annealing,…

How to get "inverse model response"?



# Differential Flatness for Distributed-Parameter Problems


**Identification of Parametric Models: from Experimental Data;** Eric Walter, and Luc Pronzato Springer; Auflage: 1997.

Conditions for a system to be differentially flat:

- Existence of a flat output

$$y^{flat} = h^{flat}(x, u, \dot{u}, \dots, u^s, p)$$

- Fulfilling the following conditions:

$$x = \Psi_x\left(y^{flat}, \dot{y}^{flat}, \dots, y^{flat^{\alpha}}, p\right)$$

$$u = \Psi_u\left(y^{flat}, \dot{y}^{flat}, \dots, y^{flat^{\alpha+1}}, p\right)$$

$$\dim y^{flat} = \dim u$$

# Case Study: Diffusion-type Problem

Considering a parabolic PDE:

$$\frac{\partial \Phi}{\partial t} = p\frac{\partial^2 \Phi}{\partial x^2} + u(x,t)$$

with $\mathrm{p} = 0.1$ and $u_0 = \sin(2\pi x)$.

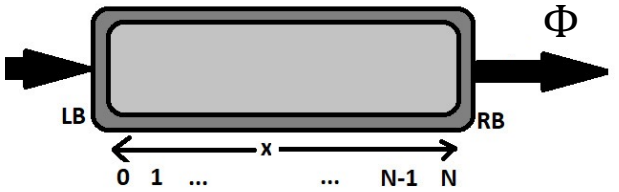



Fig. S1:A simple diffusion system.

- The system is differentially flat with a flat output defined as the vector:

$$y^{flat} = [y_{t_k,1}, y_{t_k,2}, \cdots, y_{t_k,N}]$$

where $y$ corresponds to the measured output of $\Phi$.

# Case Study: Diffusion-type Problem + Neural ODEs

Considering a parabolic PDE:

$$\frac{\partial \Phi}{\partial t} = p\frac{\partial^2 \Phi}{\partial x^2} + u(x,t)$$

with $\mathrm{p} = 0.1$ and $u_0 = \sin(2\pi x)$.

No model,
Only measurements

***An alternative approach…***

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

- Uses neural ODEs to get the data-driven model of the process under study, generally represented as:

$$\begin{cases} \dot{x}(t) = NN(x(t), u(t), p); \\ x(t_0) = x_0 \end{cases}$$

- This acts as a surrogate model – the output functions and their derivatives are derived.

- ✓ Discovers and creates numerical solutions
- ✓ Including time derivatives
- ✓ Might be combined with process knowledge (i.e., physics informed neural networks)

# Control Input by Model Inversion

- Control input (at a particular point in dimension) in time domain is calculated using the formula:

$$u(y_{t_k,1}) = \dot{y}_{1,1} - \frac{p}{\Delta x^2}\phi_{t_k,N+1} + \frac{2p}{\Delta x^2}\, y_{t_k,N} - \frac{p}{\Delta x} y_{t_k,N-1}$$

- Extending this throughout the dimensional space – new control $u_{data}$ fed back to get an effective control of diffusion.

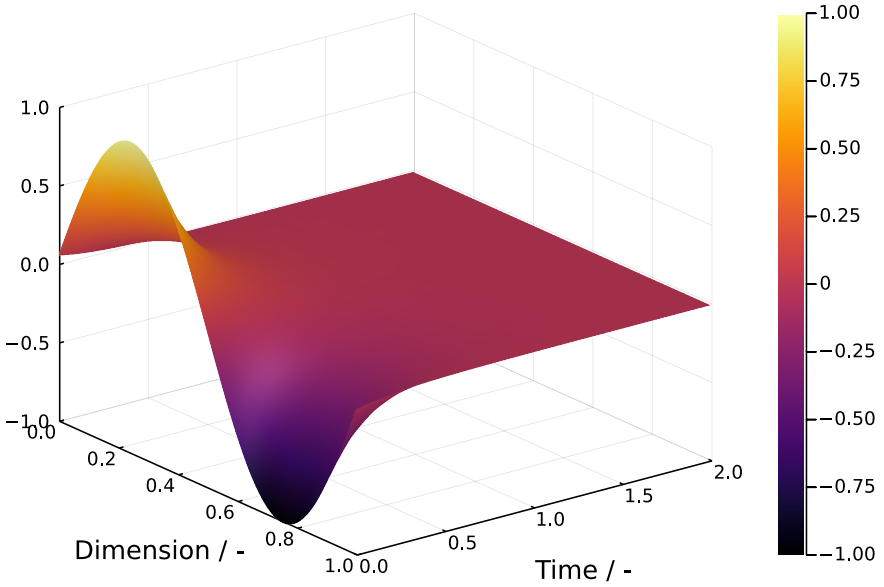



Fig. S2a: A conventional diffusion profile for the chosen diffusion coefficient $p = 0.1$.

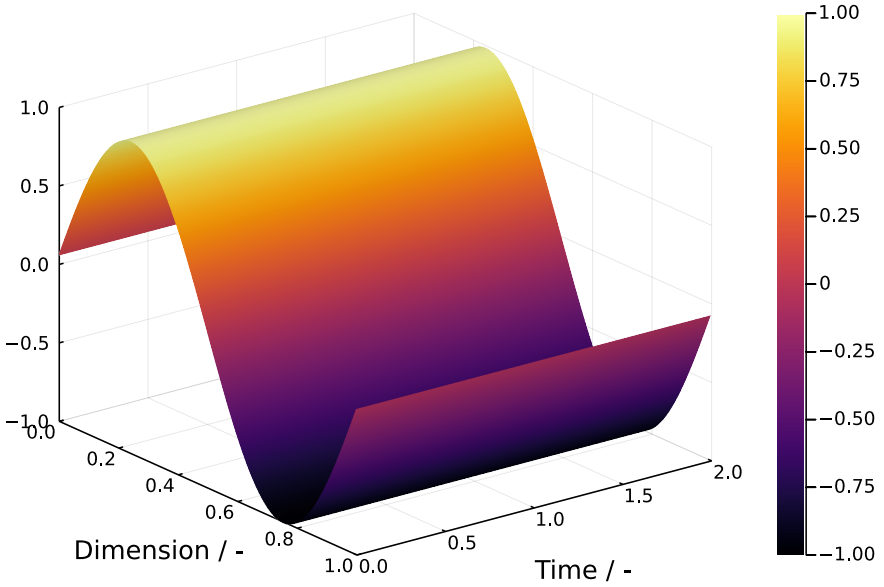
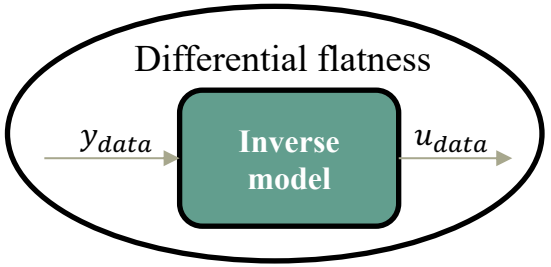
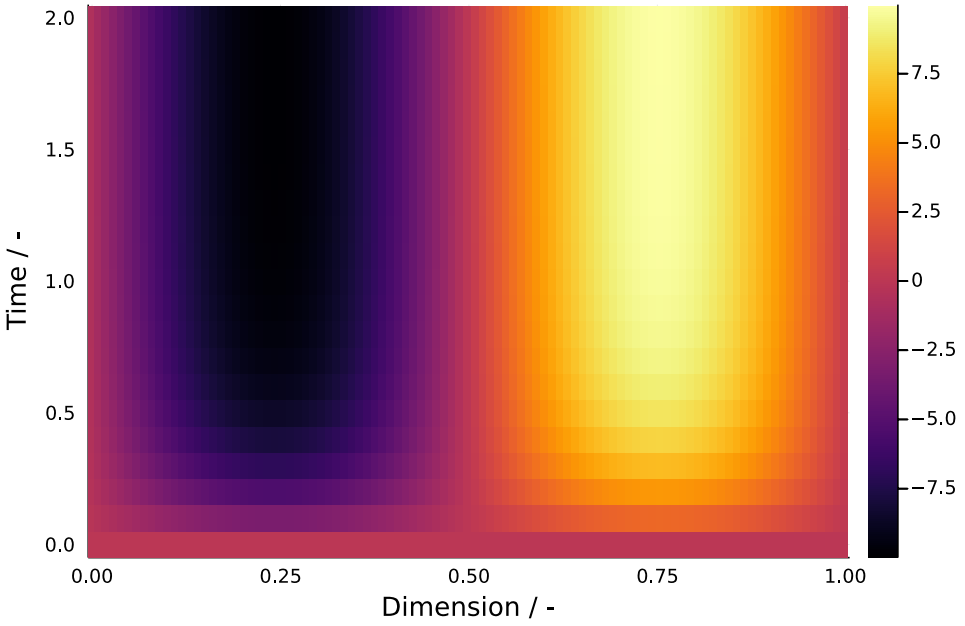


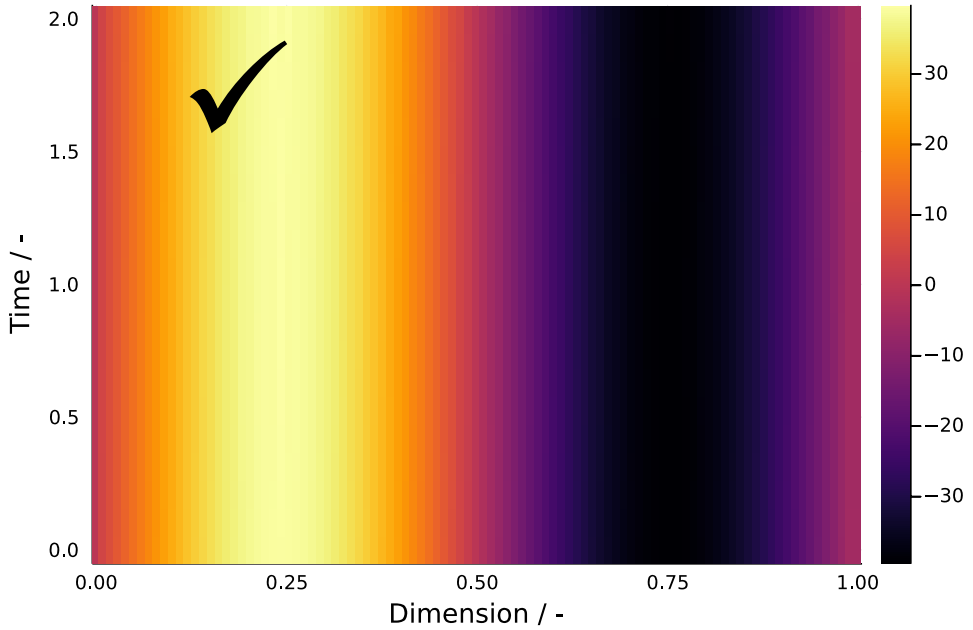

Fig. 2b: Compensated diffusion profile by closing the loop with $u_{data}$ from the inverse model.

# Parameter Sensitivity Analyses




1. $\frac{\partial y}{\partial p} :=$ parameter sensitivities based on output profiles
2. $\frac{\partial u}{\partial p} :=$ parameter sensitivities based on input profiles




Higher sensitivity values

Better parameter estimation

Fig. S3: Sensitivities to the diffusion parameter variation: (a) based on the measured output: (b) based on the re-constructed input.

# Conclusions

- Standard framework for parameter identification which gets its loss function based on the output data is given an advancement – introducing an input-based framework.
- Differential flatness concept – to calculate the control input.
- The surrogate model involving neural ODEs – to get the output functions and their derivatives.
- Sensitivity analyses – based on the output data and the control input.
- Input-based analysis – higher sensitivity value, so better probability to identify the parameter.
- But the calculated input $u_{data}$ depends on the quality of model inversion and the neural ODE surrogate model – could be improved using more optimization design tools.

▲ Hochschule Harz
Harz University of Applied Sciences

Subiksha Selvarajan
Email: sselvarajan@hs-harz.de
Friedrichstrasse 57 – 59
38855 Wernigerode
Germany