AUSTRIAN INSTITUTE OF TECHNOLOGY
TOMORROW TODAY

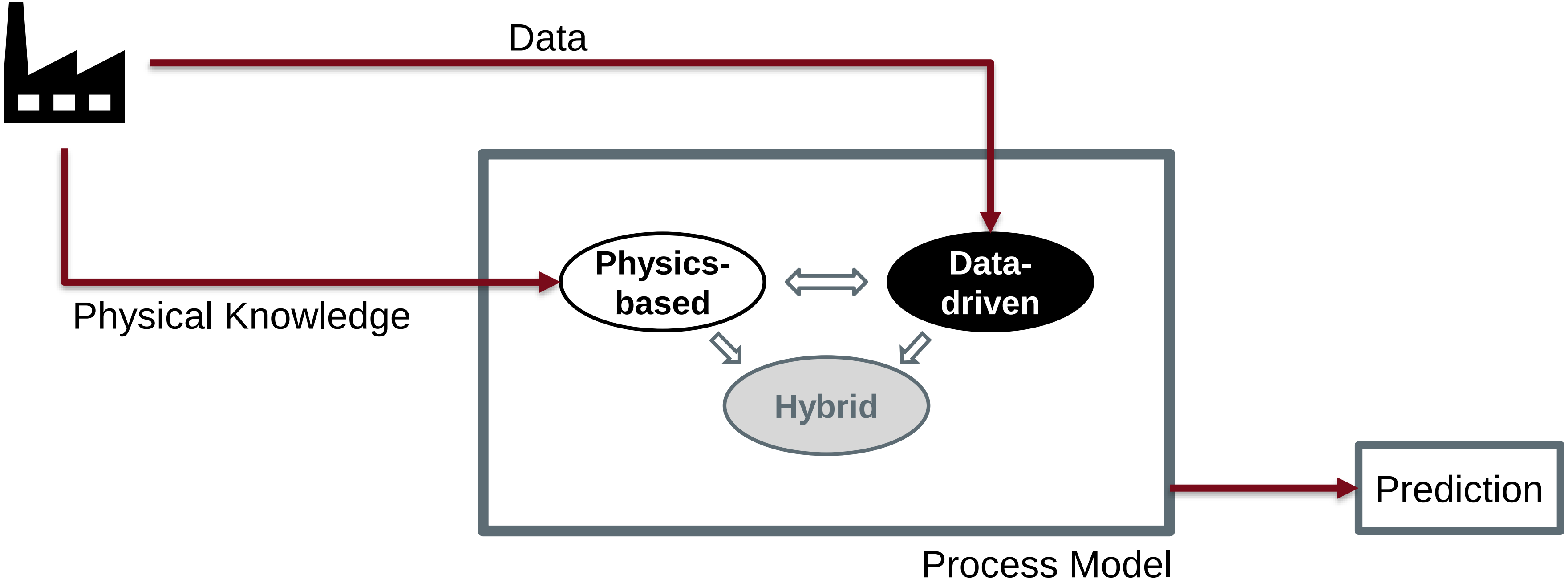
INTRODUCTION OF A PRIORI DOMAIN KNOWLEDGE IN FUNCTION APPROXIMATION USING SHAPE-CONSTRAINT P-SPLINES

Jakob Weber

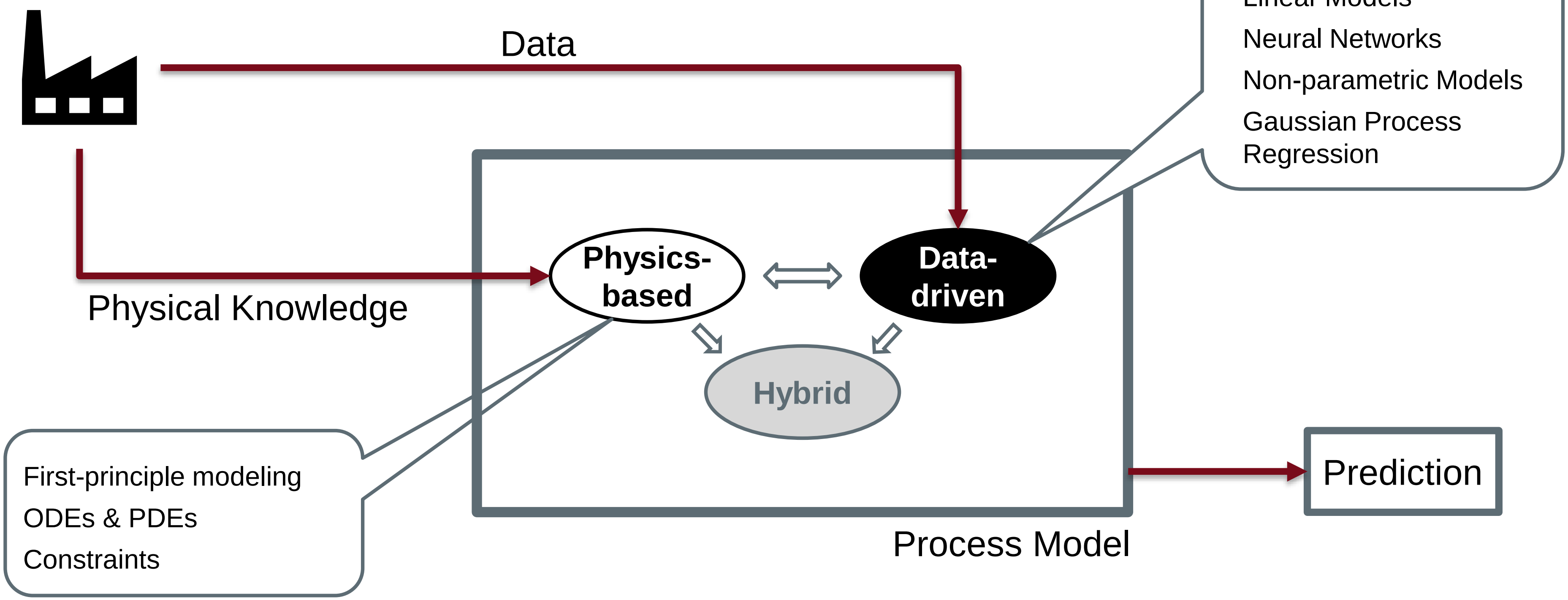
OVERVIEW - AGENDA

- Problem formulation & motivation
- Linear models, Regularization & Penalized Least-Squares
- B-splines & P-splines
- Shape-constraint P-splines
- Experiments
- Summary & Questions

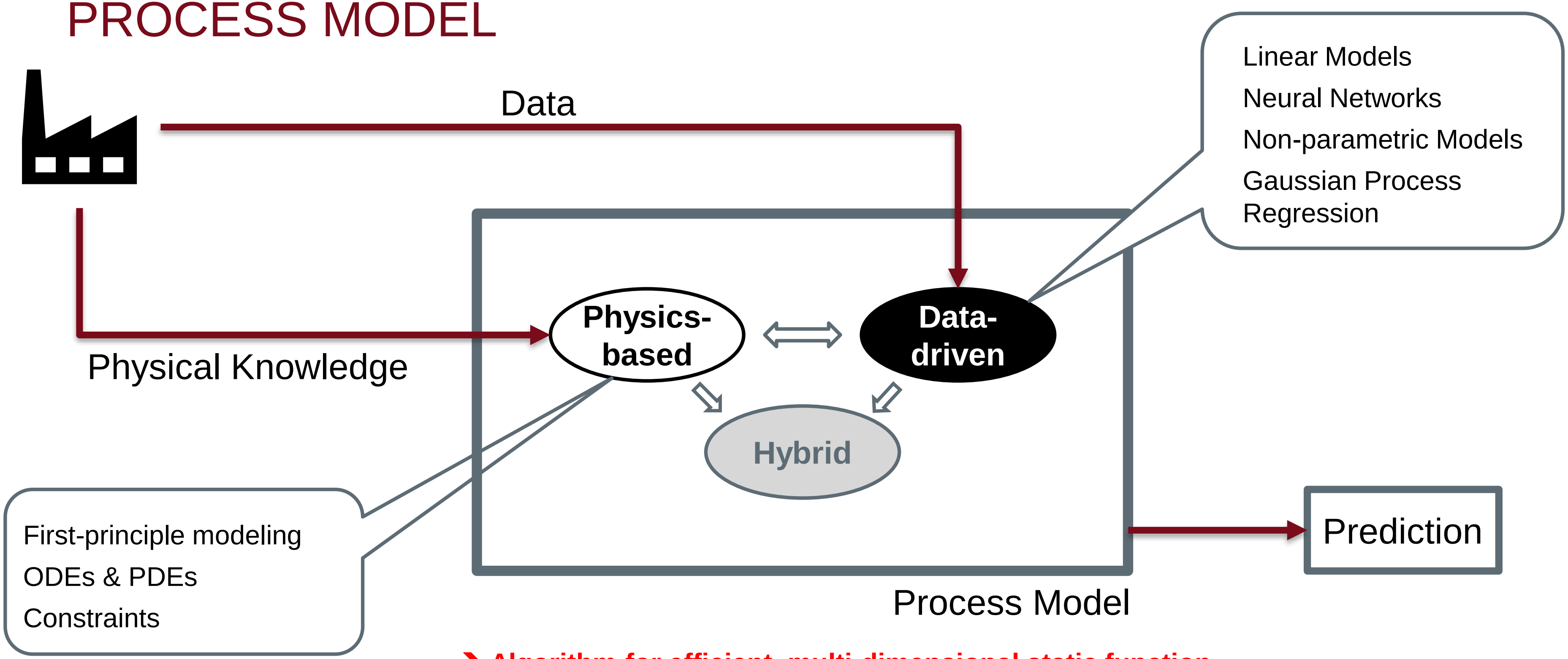
PROCESS MODEL



PROCESS MODEL



PROCESS MODEL

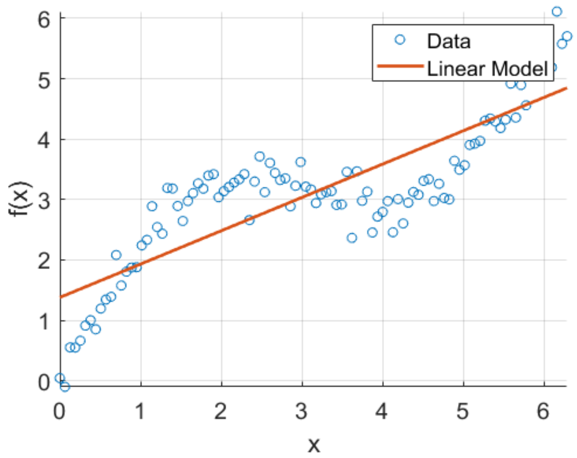


➔ Algorithm for efficient, multi-dimensional static function approximation using a priori domain knowledge

LINEAR MODELS

- $\mathcal{D} = \{(x_1^{(i)}, \ldots, x_q^{(i)}, y^{(i)}),\ i = 1, 2, \ldots, n\}$
- Model the data assuming additive noise ϵ

$$y = f(x_1, \ldots, x_q) + \epsilon$$



- Unknown function f is a linear combination of the inputs

$$f(x_1, \ldots, x_q) = \beta_0 + \beta_1 x_1 + \cdots + \beta_q x_q$$

$$f(x_1, \ldots, x_q) = \mathbf{x}^{\mathrm{T}} \beta$$

$$\mathbf{x}^{\mathrm{T}} = [1, x_1, \ldots, x_q] \in \mathbb{R}^{1 \times q+1}$$

$$\beta^{\mathrm{T}} = [\beta_0, \beta_1, \ldots, \beta_q] \in \mathbb{R}^{1 \times q+1}$$

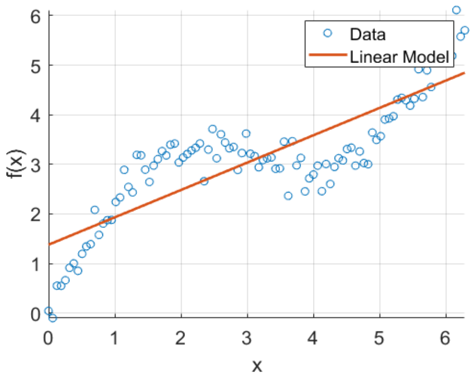
$$\epsilon^{\mathrm{T}} = [\epsilon^{(1)}, \ldots, \epsilon^{(n)}] \in \mathbb{R}^{1 \times n}$$

- And with the complete data

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

$$\mathbf{X} = \begin{bmatrix} 1 & x_1^{(1)} & \ldots & x_q^{(1)} \\ \vdots & \vdots & & \vdots \\ 1 & x_1^{(n)} & \ldots & x_q^{(n)} \end{bmatrix} \in \mathbb{R}^{n \times q+1}$$

LEAST SQUARES FORMULATION



- Linear model

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

- Least squares loss function

$$\mathrm{LS}(\mathbf{y}, \beta) = \|\mathbf{y} - \mathbf{X}\beta\|_2^2$$

$$\begin{aligned}\mathrm{LS}(\mathbf{y}, \beta) &= (\mathbf{y} - \mathbf{X}\beta)^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\beta) \\ &= \mathbf{y}^{\mathrm{T}}\mathbf{y} - 2\mathbf{y}^{\mathrm{T}}\mathbf{X}\beta + \beta^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\beta\end{aligned}$$

$$\frac{\partial \mathrm{LS}(\mathbf{y}, \beta)}{\partial \beta} = -2\mathbf{X}^{\mathrm{T}}\mathbf{y} + 2\beta^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X} = 0$$

- Least squares estimate

$$\boxed{\hat{\beta}_{\mathrm{LS}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}}$$

REGULARIZATION & PENALIZED LEAST SQUARES

- Restrict the parameter space by adding a penalty term to the loss function depending on the complexity of the model

$$\mathrm{PLS}(\mathbf{y}, \beta; \lambda) = \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \cdot \mathrm{pen}(\beta)$$

with $\mathrm{pen}(\beta) = \|\beta\|_{\mathbf{K}}^2 = \beta^{\mathrm{T}}\mathbf{K}\beta$ and positive definite penalty matrix $\mathbf{K} \in \mathbb{R}^{p \times p}$

leading to $\hat{\beta}_{\mathrm{PLS}} = \arg\min_{\beta} (\mathrm{PLS}(\mathbf{y}, \beta; \lambda)) = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{K})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$

Question: How do we cope with non-linear behavior in the data?

B-SPLINES

- Lin. comb. of d B-spline basis functions B_j^l of order l defined on a sequence of knots K

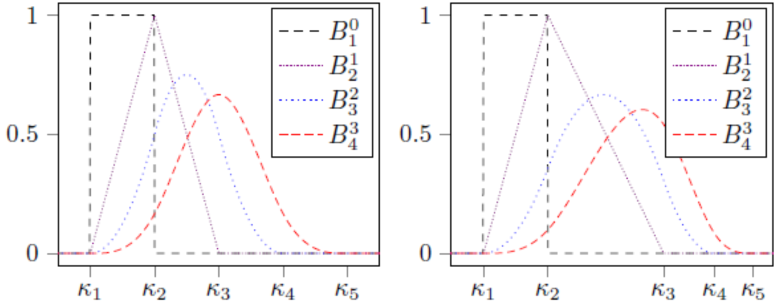
$$s(x) = \sum_{j=1}^{d} B_j^l(x)\beta_j$$

$$K = \{\kappa_{1-l}, \kappa_{1-l+1}, \dots, \kappa_{d+1}\}$$

- Each basis function B_j^l is a piecewise polynomial and defined recursively as

$$B_j^0(x) = \begin{cases} 1 & \text{for} \quad \kappa_j \leq x < \kappa_{j+1} \\ 0 & \text{otherwise} \end{cases}$$

$$B_j^l(x) = \frac{x - \kappa_{j-l}}{\kappa_j - \kappa_{j-l}} B_{j-1}^{l-1}(x) + \frac{\kappa_{j+1} - x}{\kappa_{j+1} - \kappa_{j+1-l}} B_j^{l-1}(x)$$



B-SPLINE & PARAMETER ESTIMATION

- Using data $\mathcal{D} = \{(x^{(i)}, y^{(i)}),\ i = 1, 2, \ldots, n\}$
  we can formulate a model with known structure

$$\mathbf{y} = \mathbf{B}\beta_b + \epsilon$$

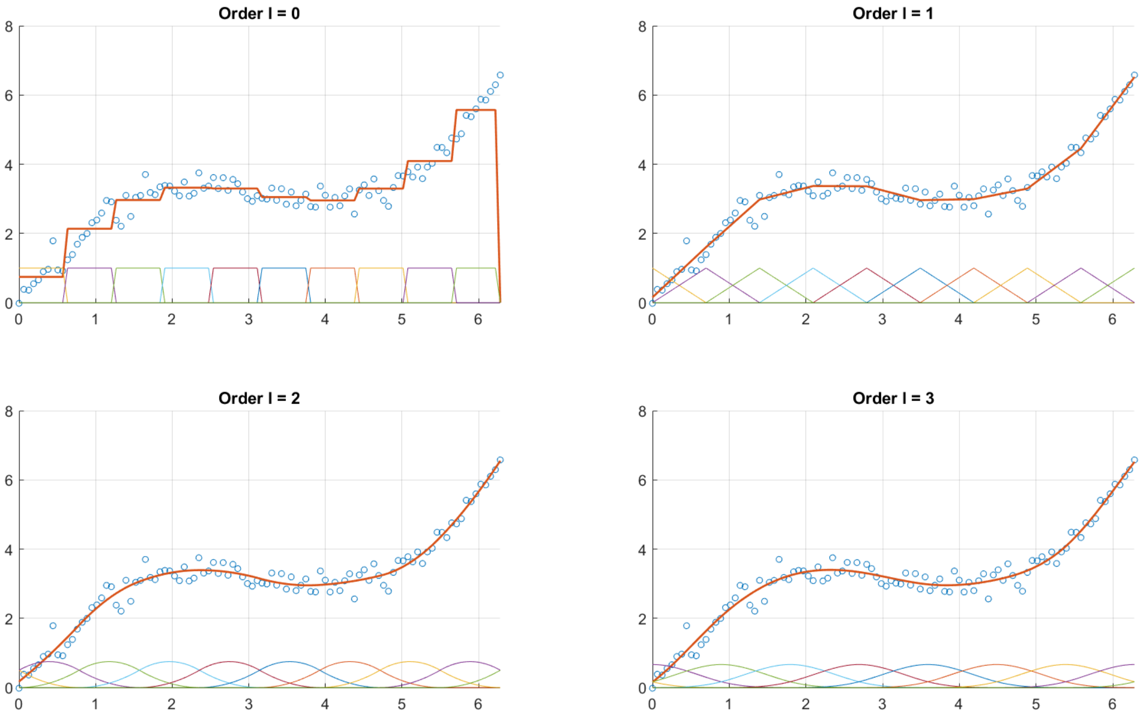
With the B-spline basis matrix

$$\mathbf{B} = \begin{bmatrix} B_1^l(x^{(1)}) & \ldots & B_d^l(x^{(1)}) \\ \vdots & & \vdots \\ B_1^l(x^{(n)}) & \ldots & B_d^l(x^{(n)}) \end{bmatrix} \in \mathbb{R}^{n \times d}$$

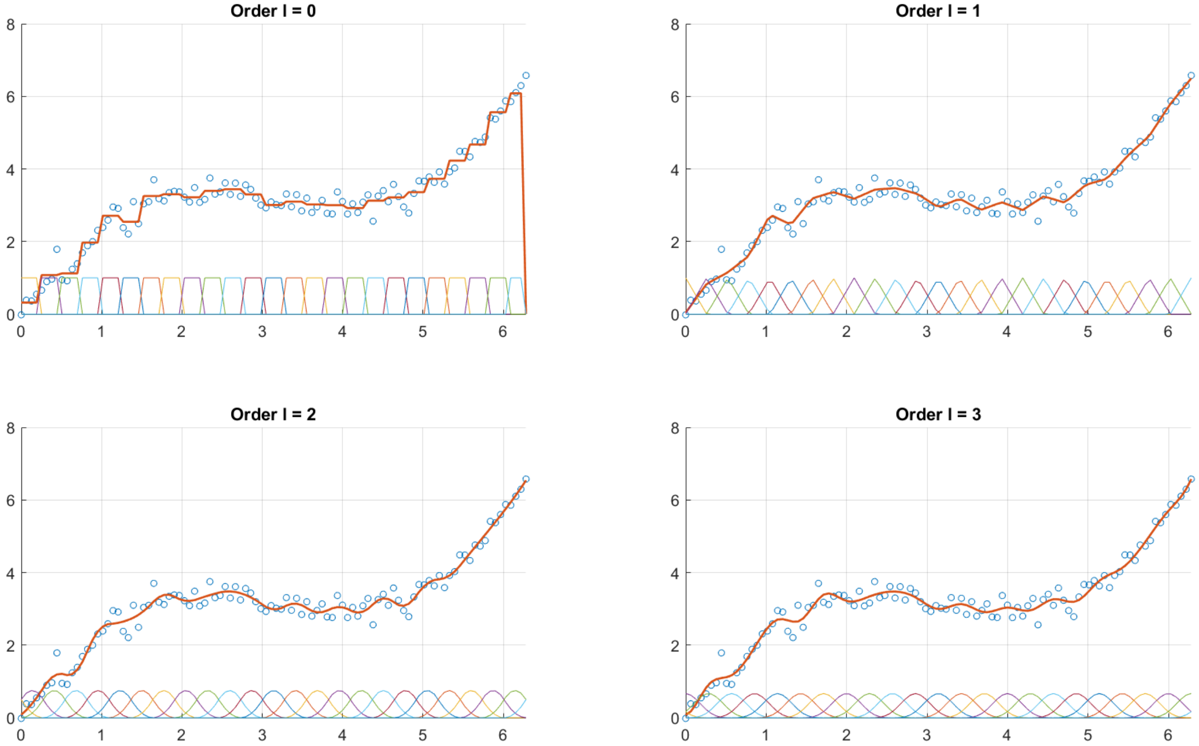
and use the least squares formulation to estimate the parameters $\boldsymbol{\beta}_b$

$$\mathrm{LS}(\mathbf{y}, \beta_b) = \|\mathbf{y} - \mathbf{B}\beta_b\|_2^2$$

B-SPLINES FOR d = 10



B-SPLINES FOR d = 25



B-SPLINE SUMMARY

- Piecewise polynomial defined on a sequence of knots
- Smoothness depends on the degree l and the number of basis functions d
- Estimate the parameters using the least squares formulation

Question: Can we use a high number of basis functions and generate a smooth estimate??

P-SPLINES

- Penalize 2nd order derivative to enforce smoothness

$$\mathrm{PLS}(\mathbf{y}, \beta; \lambda) = \|\mathbf{y} - \mathbf{B}\beta_b\|_2^2 + \lambda \int \left(f''(x)\right)^2 \, \mathrm{d}x$$

- Eilers & Marx 1996: base penalty on 2nd order finite differences of adjacent coefficients to enforce smoothness

$$\sum_{j=3}^{d} \left(\Delta^2 \beta_j\right)^2 \propto \int \left(f''(x)\right)^2 \, \mathrm{d}x$$

P-SPLINES & PLS FORMULATION

$$\sum_{j=3}^{d} \left(\Delta^2 \beta_j\right)^2 \propto \int \left(f''(x)\right)^2 \, \mathrm{d}x$$

$$\begin{aligned}\Delta^2 \beta_j &= \Delta(\Delta \beta_j) \\ &= \Delta(\beta_j - \beta_{j-1}) \\ &= \beta_j - 2\beta_{j-1} + \beta_{j-2}\end{aligned}$$

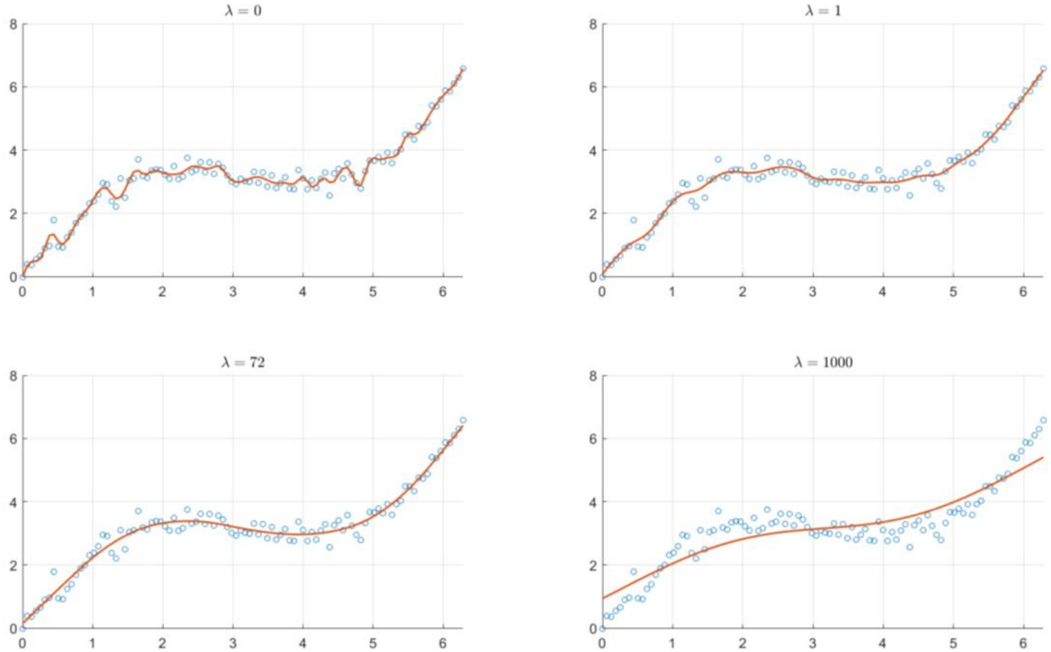
$$\lambda \sum_{j=3}^{d} \left(\Delta^2 \beta_j\right)^2 = \lambda \beta_b^{\mathrm{T}} \mathbf{D}_2^{\mathrm{T}} \mathbf{D}_2 \beta_b = \lambda \beta_b^{\mathrm{T}} \mathbf{K} \beta_b$$

$$\mathbf{D}_2 \beta_b = \begin{bmatrix} 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_d \end{bmatrix}$$

$$\boxed{\mathrm{PLS}(\mathbf{y}, \beta; \lambda) = \|\mathbf{y} - \mathbf{B}\beta_b\|_2^2 + \lambda \beta_b^{\mathrm{T}} \mathbf{K} \beta_b}$$

CHOOSING λ

- λ determines the effect of the smoothness penalty
- We choose it based on the generalized cross-validation criterium GCV



P-SPLINE SUMMARY

- Expand the B-spline formulation by a smoothness constraint based on the 2nd order finite differences of adjacent parameters
- Penalized least squares formulation for parameter estimation
- Choose optimal smoothness parameter λ based on GCV criterium

Question: Can we use P-splines and introduce a priori domain knowledge??

SHAPE-CONSTRAINT P-SPLINES

$$\sum_{j=3}^{d} \left(\Delta^2 \beta_j\right)^2 \propto \int \left(f''(x)\right)^2 \, dx$$

- Expand the idea of Eilers & Marx leading to P-splines by introducing shape constraints of the form

$$\lambda_c \cdot \mathrm{con}(\beta_b) = \lambda_c \beta_b^{\mathrm{T}} \mathbf{K}_c \beta_b \quad \text{with} \quad \mathbf{K}_c = \mathbf{D}^{\mathrm{T}} \mathbf{V}_c \mathbf{D} \ \in \mathbb{R}^{d \times d}$$

- The shape constraint matrix $\mathbf{K}$ is consists of
  - a mapping matrix $\mathbf{D}$ based on finite differences of adjacent parameters
  - a weighting matrix $\mathbf{V}_C$ determining if the constraint is active
- This leads to the following formulation

$$\mathrm{PLS}_{\mathrm{SCP}}(\mathbf{y}, \beta_b; \lambda, \lambda_c) = \|\mathbf{y} - \mathbf{B}\beta_b\|_2^2 + \lambda \beta_b^{\mathrm{T}} \mathbf{K} \beta_b + \lambda_c \beta_b^{\mathrm{T}} \mathbf{K}_c \beta_b$$

which can be solved iteratively given the algorithm on the next slide.

$$\mathrm{PLS}_{\mathrm{SCP}}(\mathbf{y}, \beta_b; \lambda, \lambda_c) = \|\mathbf{y} - \mathbf{B}\beta_b\|_2^2 + \lambda\beta_b^{\mathrm{T}}\mathbf{K}\beta_b + \lambda_c\beta_b^{\mathrm{T}}\mathbf{K}_c\beta_b$$

Algorithm 1: Estimation of the shape-constraint P-spline coefficients.

Result: $\hat{\beta}_{\mathrm{SC}}$

$i \leftarrow 1;$

$\hat{\beta}_i \leftarrow$ Solution from (3.7) for $\lambda_c = 0;$

$\mathbf{V}_c^0 \leftarrow \mathbf{0};$

$\mathbf{V}_c^1 \leftarrow \mathbf{V}_c(\hat{\beta}_i);$

while $\mathbf{V}_c^i \neq \mathbf{V}_c^{i-1}$ **do**

$\quad \hat{\beta}_{i+1} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{D}_2^{\mathrm{T}}\mathbf{D}_2 + \lambda_c\mathbf{D}_c^{\mathrm{T}}\mathbf{V}_c^i\mathbf{D}_c)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y};$

$\quad \mathbf{V}_c^{i+1} \leftarrow \mathbf{V}_c(\hat{\beta}_{i+1});$

$\quad i \leftarrow i + 1;$

end

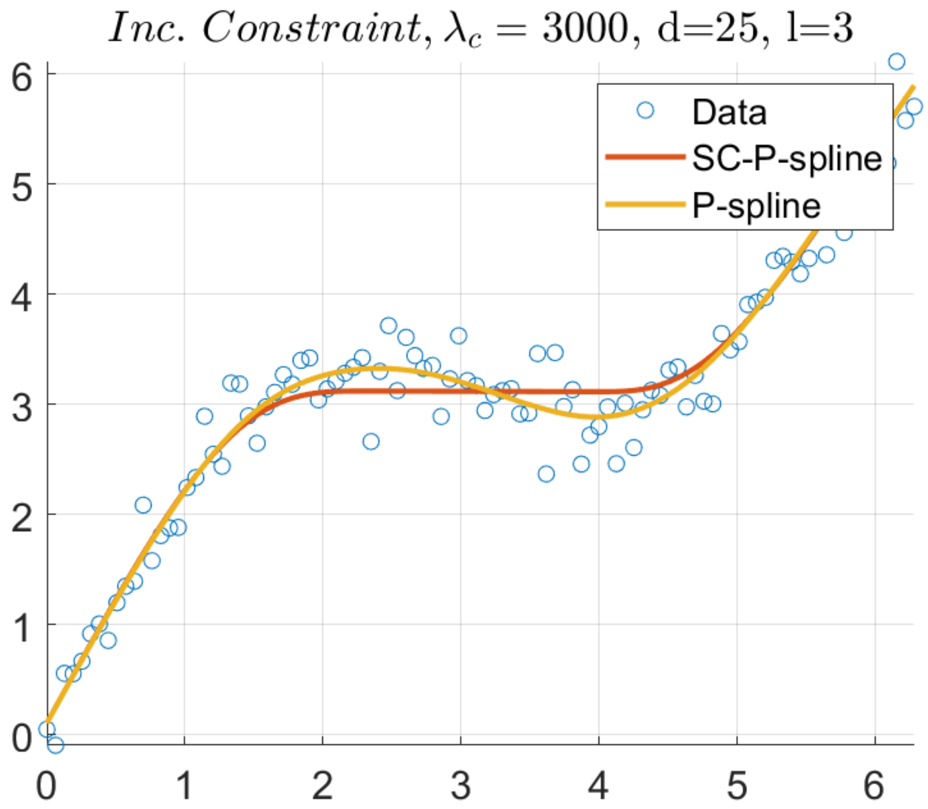
$\hat{\beta}_{\mathrm{SC}} \leftarrow \hat{\beta}_i;$

EXAMPLE INCREASING BEHAVIOUR

- Constraint formulation $$\lambda_c \int (f(x)')^2 \, \mathrm{d}x \quad \text{if } f'(x) < 0$$

- Mapping matrix $$\mathbf{D}_1 \beta_b = \begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_d \end{bmatrix}$$

- Weighting matrix $$v_{j-1}(\beta_b) = \begin{cases} 0, & \text{if } \Delta^1 \beta_j \geq 0 \\ 1, & \text{if } \Delta^1 \beta_j < 0 \end{cases} \quad \text{for } j = 2, 3, \ldots, d$$



$Inc.\ Constraint, \lambda_c = 3000$, d=25, l=3
Data
SC-P-spline
P-spline
0
1
2
3
4
5
6
0
1
2
3
4
5
6

VARIOUS CONSTRAINTS

- Through the choice of the mapping matrix D and the weighting matrix V various constraints can be incorporated

Constraint		Description
Jamming		$f(x^{(p)}) \approx y^{(p)}$
Boundedness	lower	$f(x) \geq B$
	upper	$f(x) \leq B$
Monotonicity	increasing	$f'(x) \geq 0$
	decreasing	$f'(x) \leq 0$
Curvature	convex	$f''(x) \geq 0$
	concave	$f''(x) \leq 0$
Unimodality	peak	$m = \arg\max_x f(x)$ $f'(x) \geq 0 \quad \text{if } x < m$ $f'(x) \leq 0 \quad \text{if } x > m$
	valley	$m = \arg\min_x f(x)$ $f'(x) \leq 0 \quad \text{if } x < m$ $f'(x) \geq 0 \quad \text{if } x > m$

EXTENSION TO HIGHER DIMENSIONS 1

- Tensor-product B-splines for 2d
  - TP B-spline basis function $$T_{j,r}(x_1, x_2) = B_j^{l_1}(x_1) B_r^{l_2}(x_2)$$
  - TP B-spline basis $$t(x_1, x_2) = \sum_{j=1}^{d_1} \sum_{r=1}^{d_2} T_{j,r}(x_1, x_2)\beta_{j,r}$$
  - TP B-spline basis matrix $$\mathbf{T} = \begin{bmatrix} T_{1,1}(x_1^{(1)}, x_2^{(1)}) & \dots & T_{d_1,d_2}(x_1^{(1)}, x_2^{(1)}) \\ \vdots & & \vdots \\ T_{1,1}(x_1^{(n)}, x_2^{(n)}) & \dots & T_{d_1,d_2}(x_1^{(n)}, x_2^{(n)}) \end{bmatrix} \in \mathbb{R}^{n \times d_1 d_2}$$
  - Least squares formulation $$\mathrm{LS}(\mathbf{y}, \beta_t) = \|\mathbf{y} - \mathbf{T}\beta_t\|_2^2$$

TENSOR-PRODUCT P-SPLINE

- Penalize 2nd order derivative in both dimensions to enforce smoothness

$$\left[\sum_{j=3}^{d_1}\sum_{r=1}^{d_2}(\Delta_1^2\beta_{j,r})^2 + \sum_{j=1}^{d_1}\sum_{r=3}^{d_2}(\Delta_2^2\beta_{j,r})^2\right] \propto \iint \left(f''(x_1,x_2)\right)^2 \, \mathrm{d}x_1 \, \mathrm{d}x_2$$

- Penalized Least Squares formulation

$$\mathrm{PLS}(\mathbf{y}, \boldsymbol{\beta}_t; \lambda) = \|\mathbf{y} - \mathbf{T}\boldsymbol{\beta}_t\|_2^2 + \lambda\boldsymbol{\beta}_t^{\mathrm{T}}\mathbf{K}\boldsymbol{\beta}_t$$

with the penalty matrix

$$\mathbf{K} = \left[(\mathbf{I}_{d_2} \otimes \mathbf{D}_{1,2})^{\mathrm{T}}(\mathbf{I}_{d_2} \otimes \mathbf{D}_{1,2}) + (\mathbf{D}_{2,2} \otimes \mathbf{I}_{d_1})^{\mathrm{T}}(\mathbf{D}_{2,2} \otimes \mathbf{I}_{d_1})\right] \in \mathbb{R}^{d_1 d_2 \times d_1 d_2}$$

TENSOR-PRODUCT SC-P-SPLINE

- Constraints available for
  - One dimension: e.g. increasing only in dimension 1
  - Both dimensions: e.g increasing for dimension 1 and decreasing for dimension 2
- Example: Increasing behavior in dimension 1

$$\mathbf{D}_c\beta_t = (\mathbf{I}_{d_2} \otimes \mathbf{D}_1)\beta_t \quad v_{j+(i-1)(d_1-1)}(\beta_t) = \begin{cases} 0, & \text{if } \Delta^1_{d_1}\beta_{j+(i-1)d_1+1} \geq 0 \\ 1, & \text{if } \Delta^1_{d_1}\beta_{j+(i-1)d_1+1} < 0 \end{cases},$$

$$\text{for } j = 1, 2, \ldots, d_1 - 1 \text{ and } i = 1, 2, \ldots, d_2$$

$$\mathrm{PLS}_{\text{SC-TP}}(\mathbf{y}, \beta_t; \lambda, \lambda_c) = \|\mathbf{y} - \mathbf{T}\beta_t\|_2^2 + \lambda\beta_t^{\mathrm{T}}\mathbf{K}\beta_t + \lambda_c\beta_t^{\mathrm{T}}\mathbf{K}_c\beta_t$$

EXTENSION TO HIGHER DIMENSIONS 2

- Additive models: from 2-d to q-d without interaction terms

$$f(x_1, \ldots, x_q) = f_1(x_1) + \cdots + f_q(x_q)$$

- Modeling each function $f(x_i)$ using a B-spline $f_i(\mathbf{x}_i) = \mathbf{B}_i \beta_{b_i}$

$$\mathbf{y} = \mathbf{B}_1 \beta_{b_1} + \cdots + \mathbf{B}_q \beta_{b_q} + \epsilon$$

$$\mathbf{y} = \mathbf{X}\beta + \epsilon = \left[\mathbf{B}_1, \ldots, \mathbf{B}_q\right] \begin{bmatrix} \beta_{b_1} \\ \vdots \\ \beta_{b_q} \end{bmatrix} + \epsilon$$

EXTENSION TO HIGHER DIMENSIONS 3

- Additive models: from 2-d to q-d with interaction terms

$$f(x_1,\ldots,x_q) = f_1(x_1) + \cdots + f_q(x_q) + f_{1,2}(x_1,x_2) + \cdots + f_{q-1,q}(x_{q-1},x_q)$$

- Modeling each $f(x_i)$ using a B-spline and each $f(x_j, x_r)$ using a TP B-spline

$$\mathbf{y} = \mathbf{B}_1\beta_{b_1} + \cdots + \mathbf{B}_q\beta_{b_q} + \sum_{j=1}^{q-1}\sum_{r>j}^{q} \mathbf{T}_{j,r}\beta_{t_{j,r}} + \epsilon$$

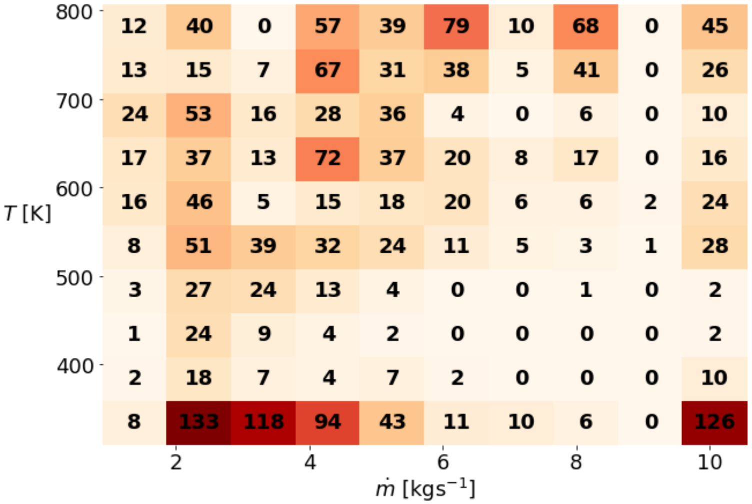
$$\mathbf{y} = \mathbf{X}\beta + \epsilon = \left[\mathbf{B}_1,\ldots,\mathbf{B}_q,\mathbf{T}_{1,2},\ldots,\mathbf{T}_{q-1,q}\right] \begin{bmatrix} \beta_{b_1} \\ \vdots \\ \beta_{b_q} \\ \beta_{t_{1,2}} \\ \vdots \\ \beta_{t_{q-1,q}} \end{bmatrix} + \epsilon$$

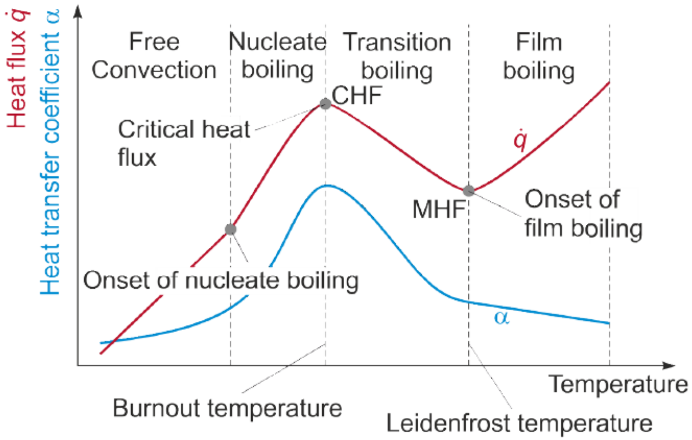
SUMMARY SC-P-SPLINES

- Expand the P-spline formulation by using a priori domain knowledge via shape constraints based on mapping matrices $\mathbf{D}$ and weighting matrices $\mathbf{V}_c$
- Iterative algorithm for parameter estimation
- Choose constraint parameter λ_C based on your trust in the a priori domain knowledge

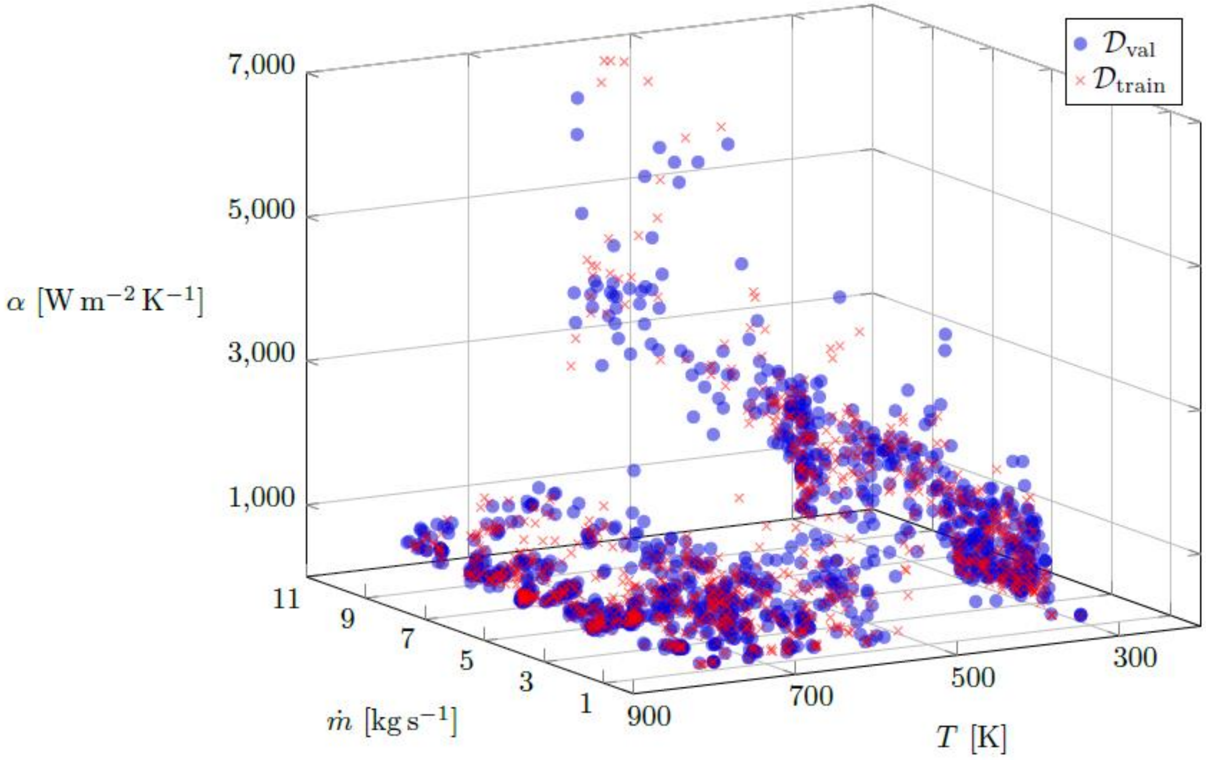
EXPERIMENT HEAT TREATMENT PROCESS

- Estimate the heat transfer coefficient $\alpha(T, \dot{m})$ based on measurements of the temperature T and the mass flow $\dot{m}$
- A priori knowledge through the Nukiyama curve (peak)
- Challenges
  - Sparse Data
  - Very noisy



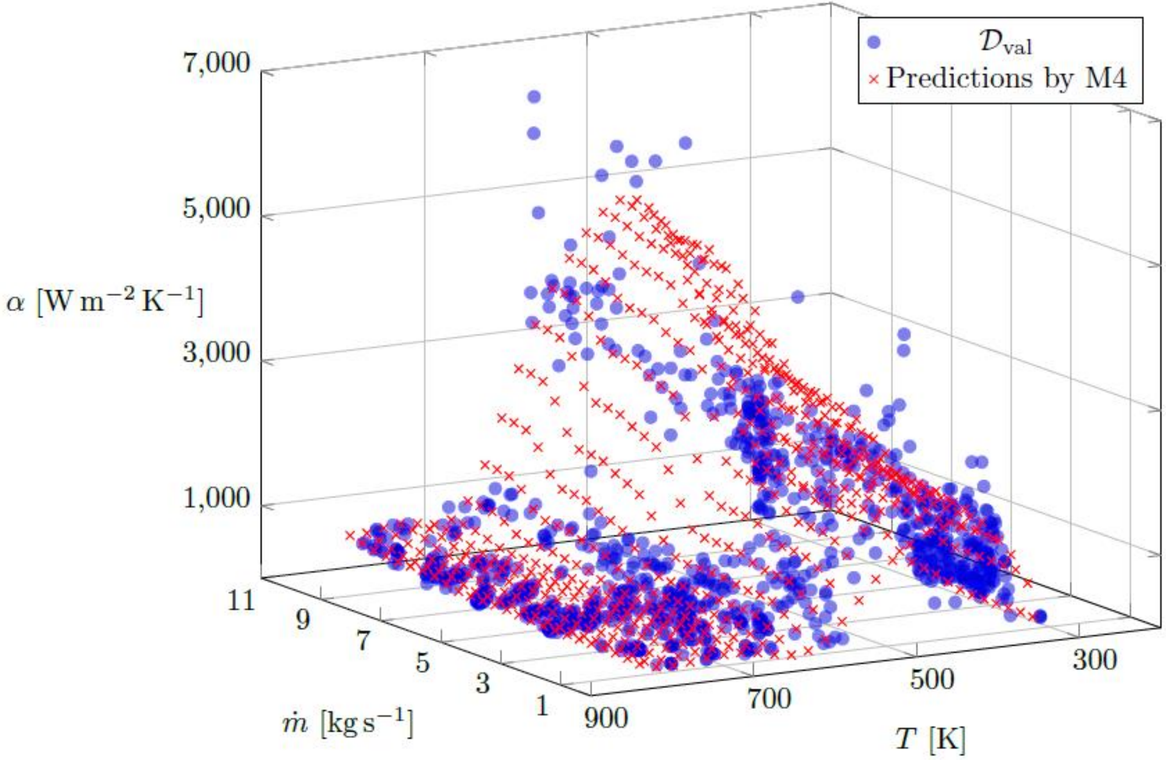


DATA


$\mathcal{D}_{\mathrm{val}}$
$\mathcal{D}_{\mathrm{train}}$
7,000
5,000
3,000
1,000
α [W m^{-2} K^{-1}]
11
9
7
5
3
1
$\dot{m}$ [kg s^{-1}]
900
700
500
300
T [K]

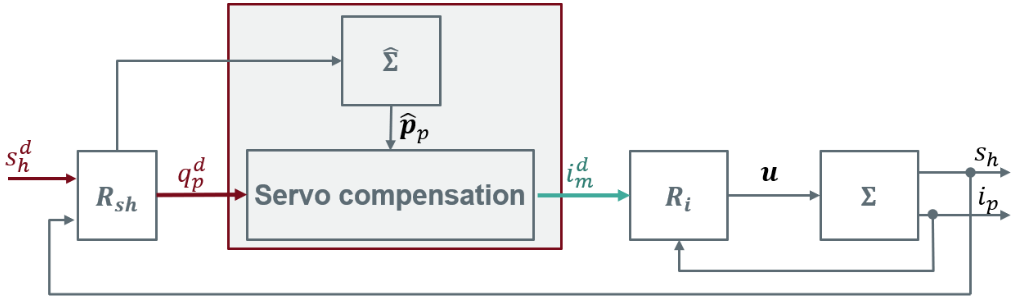
BEST MODEL

$\mathrm{M4} = t(\dot{m}, T)$, with monotonicity for input $\dot{m}$



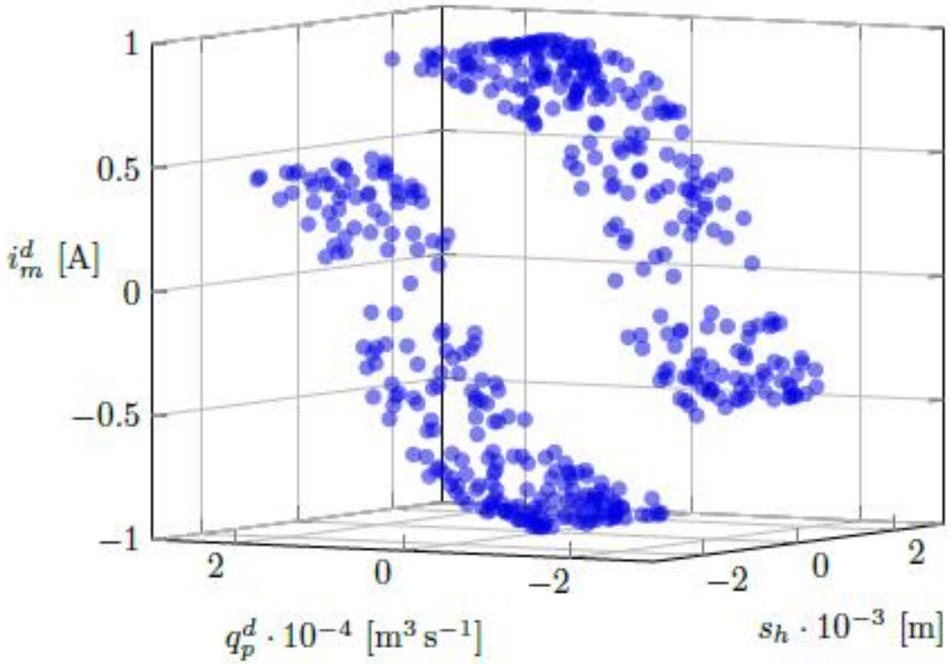
EXPERIMENT HYBRID MODELING AND CONTROL

- Estimate the differential current $i_m^d(q_p^d, s_h^d)$ based on measurements of the desired main valve position s_h^d and the desired differential flow q_p^d
- Block diagram is given on the right



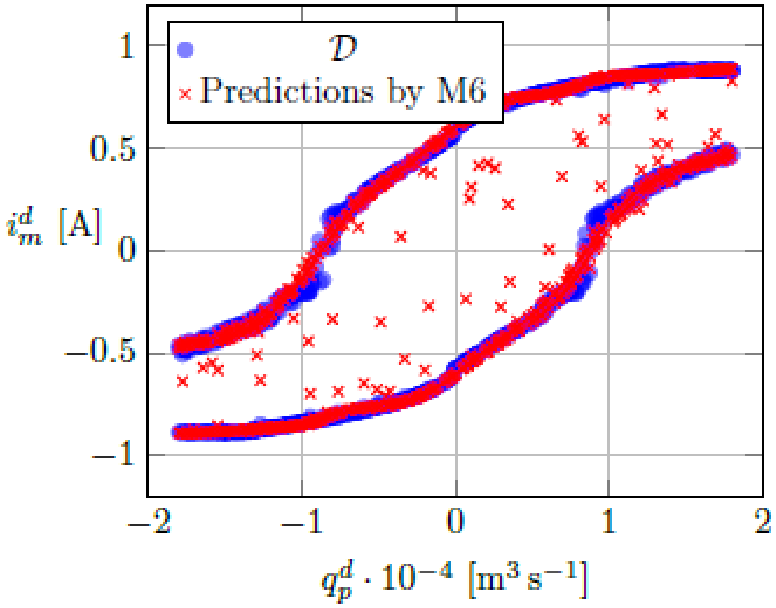
- Challenges
  - Discontinuous data

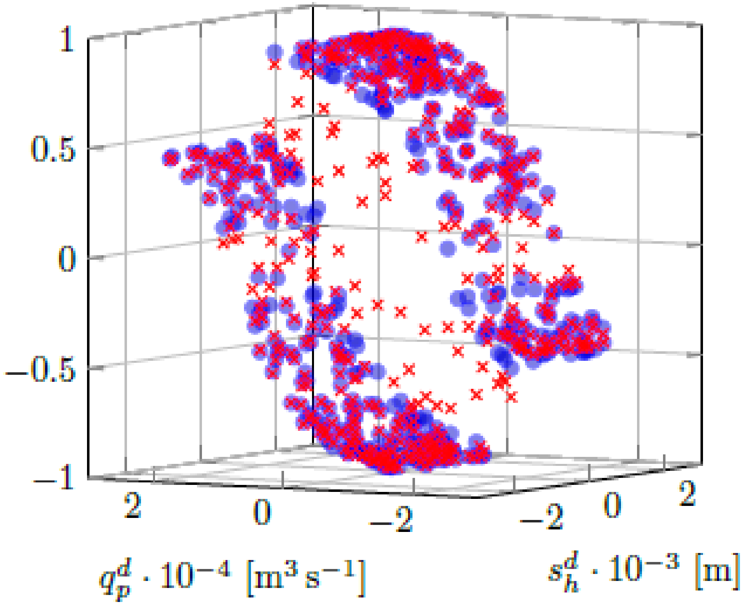
DATA



BEST MODEL

$\text{M6} = s\left(s_h^d\right) + t\left(q_p^d, s_h^d\right)$, with monotonicity for both dimensions





SUMMARY

- Linear models & theory for function approximation
- Extension to nonlinear functions approximation using B-splines
- Optimal smoothness using P-splines
- Incorporation of a priori domain knowledge through shape-constraint P-splines
- Multi-dim through additive models

THANK YOU!

QUESTIONS?

Jakob Weber