Universität Rostock

Traditio et Innovatio

CoM

Chair of Mechatronics

Interval-Based Model-Predictive Control for Uncertain Dynamic Systems with Actuator Constraints

SWIM 2012: Small Workshop on Interval Methods

Oldenburg, Germany

06 June, 2012

Andreas Rauh, Julia Kersten, Thomas Dötschel, Harald Aschemann

Chair of Mechatronics
University of Rostock, Germany

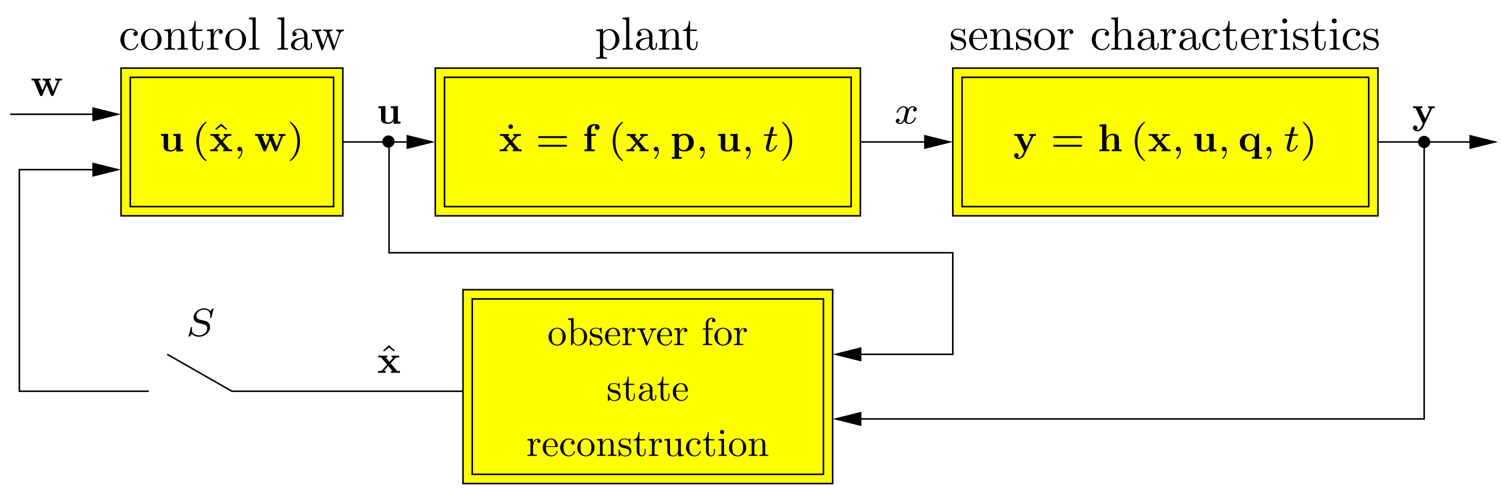
Contents

- Tracking control and stabilization of desired operating points for control systems with uncertainties
- Different control methodologies
  - Feedback linearizing control laws
  - Exploitation of differential flatness
  - Sliding mode control
  - Classical model-predictive control
- Model-predictive control for uncertain systems
- Illustrative example: Trajectory tracking, overshoot prevention, path following
- Model-predictive control for SOFC models with uncertainties
- Detection of overestimation in interval-based predictive control laws
- Conclusions and outlook

Tracking Control for Continuous-Time Dynamical Systems

Consider a dynamical system with

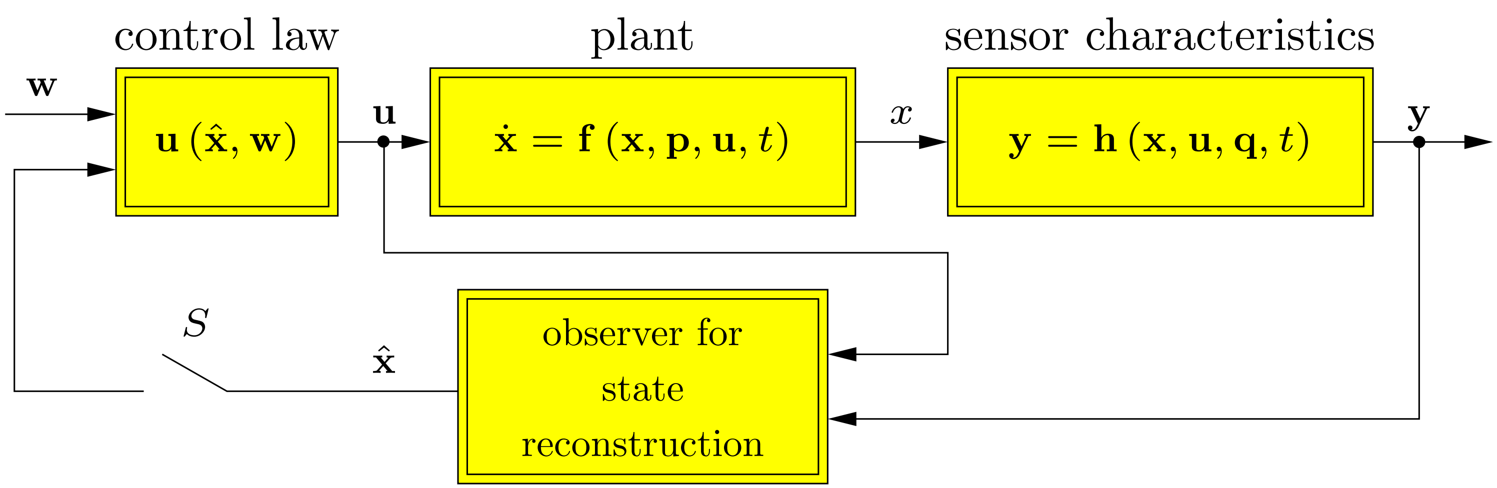
- the state equations $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{p}(t), \mathbf{u}(t), t)$
- the output $\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t))$, for example, measured data $\mathbf{h}(\cdot)$
- the desired output trajectory $\mathbf{y}_d(t)$



Tracking Control for Continuous-Time Dynamical Systems

Consider a dynamical system with

- the state equations $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{p}(t), \mathbf{u}(t), t)$
- the output $\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t))$, for example, measured data $\mathbf{h}(\cdot)$
- the desired output trajectory $\mathbf{y}_d(t)$



Necessity for state/ output feedback to prevent the violation of feasibility constraints in the case of parameter uncertainties as well as measurement and state reconstruction errors.

Tracking Control for Differentially Flat Systems

Differential Flatness of $Nonlinear$ **Dynamical Systems** $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$

A dynamical system is called differentially flat, if flat outputs

$$\mathbf{y} = \mathbf{y}\left(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \ldots, \mathbf{u}^{(\alpha)}\right)$$

exist such that

(i) all system states $\mathbf{x}$ and all inputs $\mathbf{u}$ can be expressed as functions of the flat outputs and their time derivatives:

$$\mathbf{x} = \mathbf{x}\left(\mathbf{y}, \dot{\mathbf{y}}, \ldots, \mathbf{y}^{(\beta)}\right) \quad \text{and} \quad \mathbf{u} = \mathbf{u}\left(\mathbf{y}, \dot{\mathbf{y}}, \ldots, \mathbf{y}^{(\beta+1)}\right)$$

(ii) the flat outputs $\mathbf{y}$ are differentially independent, i.e., they are not coupled by differential equations.

Note:

(a) If (i) is fulfilled, (ii) is equivalent to $\dim(\mathbf{u}) = \dim(\mathbf{y})$.

(b) The flat outputs $\mathbf{y}$ need not be the physical outputs of the dynamical system.

(c) For linear systems, differential flatness is equivalent to controllability.

Tracking Control for Differentially Flat Systems

Differential Flatness of *Nonlinear* Dynamical Systems $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t))$

A dynamical system is called differentially flat, if flat outputs

$$\mathbf{y} = \mathbf{y}\left(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \ldots, \mathbf{u}^{(\alpha)}\right)$$

exist such that

(i) all system states $\mathbf{x}$ and all inputs $\mathbf{u}$ can be expressed as functions of the flat outputs and their time derivatives:

$$\mathbf{x} = \mathbf{x}\left(\mathbf{y}, \dot{\mathbf{y}}, \ldots, \mathbf{y}^{(\beta)}\right) \quad \text{and} \quad \mathbf{u} = \mathbf{u}\left(\mathbf{y}, \dot{\mathbf{y}}, \ldots, \mathbf{y}^{(\beta+1)}\right)$$

(ii) the flat outputs $\mathbf{y}$ are differentially independent, i.e., they are not coupled by differential equations.

One possibility to solve the tracking control task is by specifying the desired system output as a time-dependent algebraic constraint to a set of ordinary differential equations or differential-algebraic equations.

Tracking Control for Differentially Flat Systems

- Guaranteed stabilization of the error dynamics by interval evaluation of suitable Lyapunov functions to account for uncertainties
- Transformation of the state equations into nonlinear controller normal form: overcompensation of uncertainties
- Sliding mode control procedures, e.g. evaluated by means of interval analysis: see previous presentation
- Alternatively: Exploitation of inherent robustness properties of model-predictive control procedures

Tracking Control for Differentially Flat Systems

- Guaranteed stabilization of the error dynamics by interval evaluation of suitable Lyapunov functions to account for uncertainties
- Transformation of the state equations into nonlinear controller normal form: overcompensation of uncertainties
- Sliding mode control procedures, e.g. evaluated by means of interval analysis: see previous presentation
- Alternatively: Exploitation of inherent robustness properties of model-predictive control procedures

(Interval-based) Predictive control approaches do not require an analytic reformulation of the state equations into a nonlinear controller normal form or into an input-affine system representation.

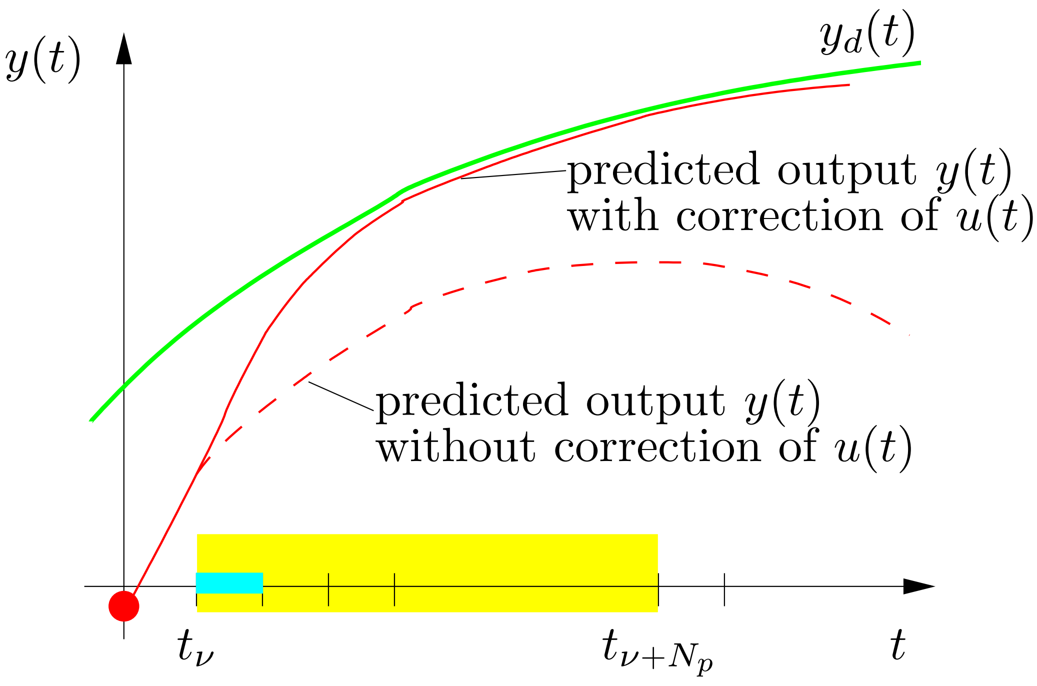
Tracking Control for Differentially Flat Systems

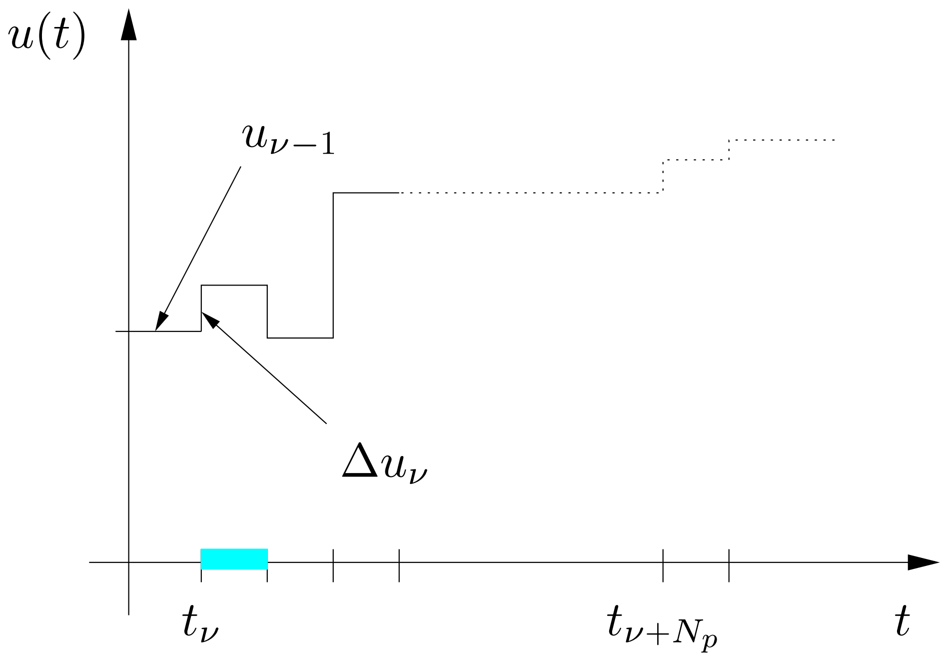
- Guaranteed stabilization of the error dynamics by interval evaluation of suitable Lyapunov functions to account for uncertainties
- Transformation of the state equations into nonlinear controller normal form: overcompensation of uncertainties
- Sliding mode control procedures, e.g. evaluated by means of interval analysis: see previous presentation
- Alternatively: Exploitation of inherent robustness properties of model-predictive control procedures

(Interval-based) Predictive control approaches do not require an analytic reformulation of the state equations into a nonlinear controller normal form or into an input-affine system representation.

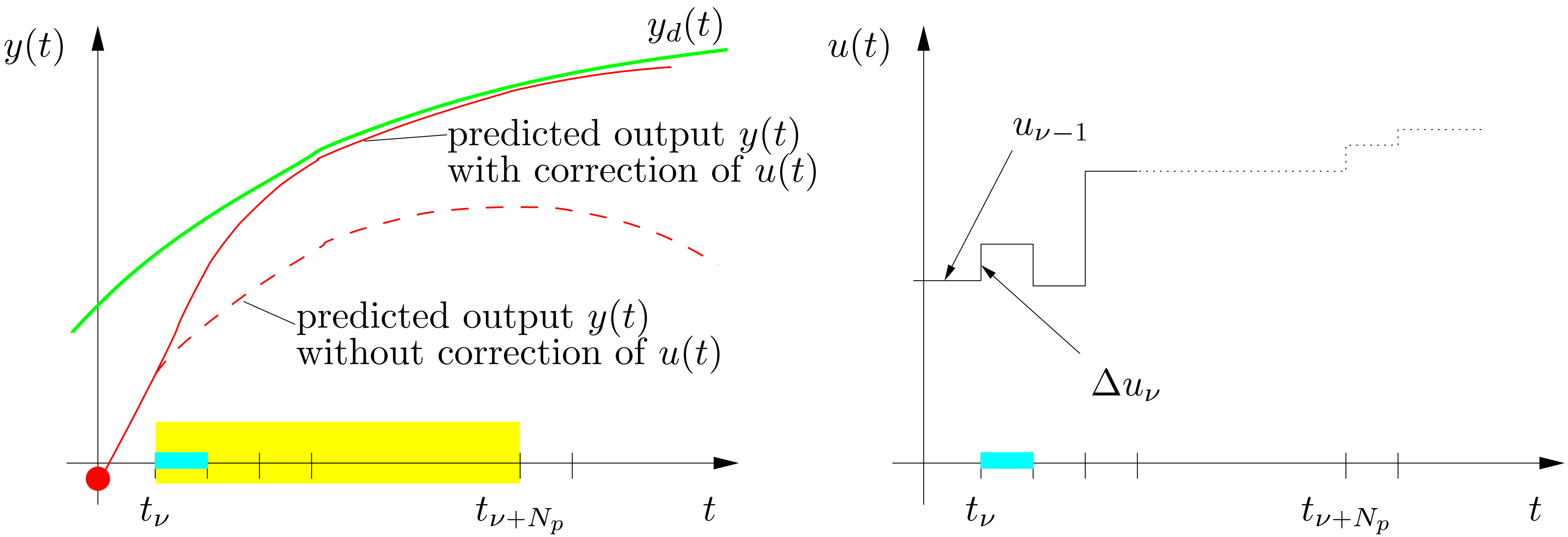
The usage of algorithmic differentiation allows for direct treatment of nonlinear system models.

Sensitivity-Based Model-Predictive Control





Sensitivity-Based Model-Predictive Control



- Sensitivity analysis for both analysis and design of control laws
- Consider a finite-dimensional dynamical system $\dot{\mathbf{x}}\left(t\right) = \mathbf{f}\left(\mathbf{x}\left(t\right), \xi\right)$ with the state vector $\mathbf{x} \in \mathbb{R}^{n_x}$ (including observer state variables) and the parameter vector $\xi \in \mathbb{R}^{n_\xi}$ (including the system parameters $\mathbf{p}$ and the control inputs $\mathbf{u}$)

Compute piecewise constant control inputs $\mathbf{u}\left(t\right)$ for each time interval $t \in [t_\nu \; ; \; t_{\nu+1})$, $0 \leq t_\nu < t_{\nu+1}$.

Sensitivity Analysis of Dynamical Systems

- Sensitivity of the solution $\mathbf{x}(t)$ to the set of ordinary differential equations $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \xi)$ with respect to a **time-invariant parameter vector** ξ

$$\frac{d}{dt}\left(\frac{\partial \mathbf{x}(t)}{\partial \xi_i}\right) = \frac{\partial \mathbf{f}(\mathbf{x}(t), \xi)}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{x}(t)}{\partial \xi_i} + \frac{\partial \mathbf{f}(\mathbf{x}(t), \xi)}{\partial \xi_i}$$

Sensitivity Analysis of Dynamical Systems

- Sensitivity of the solution $\mathbf{x}(t)$ to the set of ordinary differential equations $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \xi)$ with respect to a **time-invariant parameter vector** ξ

$$\frac{d}{dt}\left(\frac{\partial \mathbf{x}(t)}{\partial \xi_i}\right) = \frac{\partial \mathbf{f}(\mathbf{x}(t), \xi)}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{x}(t)}{\partial \xi_i} + \frac{\partial \mathbf{f}(\mathbf{x}(t), \xi)}{\partial \xi_i}$$

- New state vectors ($\mathbf{x} \in \mathbb{R}^{n_x}$, $\xi \in \mathbb{R}^{n_\xi}$)

$$\mathbf{s}_i(t) := \frac{\partial \mathbf{x}(t)}{\partial \xi_i} \in \mathbb{R}^{n_x} \quad \text{for all} \quad i = 1, \ldots, n_\xi$$

$$\dot{\mathbf{s}}_i(t) = \frac{\partial \mathbf{f}(\mathbf{x}(t), \xi)}{\partial \mathbf{x}} \cdot \mathbf{s}_i(t) + \frac{\partial \mathbf{f}(\mathbf{x}(t), \xi)}{\partial \xi_i}$$

Sensitivity Analysis of Dynamical Systems

- Sensitivity of the solution $\mathbf{x}(t)$ to the set of ordinary differential equations $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \xi)$ with respect to a **time-invariant parameter vector** ξ

$$\frac{d}{dt}\left(\frac{\partial \mathbf{x}(t)}{\partial \xi_i}\right) = \frac{\partial \mathbf{f}(\mathbf{x}(t), \xi)}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{x}(t)}{\partial \xi_i} + \frac{\partial \mathbf{f}(\mathbf{x}(t), \xi)}{\partial \xi_i}$$

- New state vectors ($\mathbf{x} \in \mathbb{R}^{n_x}$, $\xi \in \mathbb{R}^{n_\xi}$)

$$\mathbf{s}_i(t) := \frac{\partial \mathbf{x}(t)}{\partial \xi_i} \in \mathbb{R}^{n_x} \quad \text{for all} \quad i = 1, \ldots, n_\xi$$

$$\dot{\mathbf{s}}_i(t) = \frac{\partial \mathbf{f}(\mathbf{x}(t), \xi)}{\partial \mathbf{x}} \cdot \mathbf{s}_i(t) + \frac{\partial \mathbf{f}(\mathbf{x}(t), \xi)}{\partial \xi_i}$$

- Initial conditions

$$\mathbf{s}_i(0) = \frac{\partial \mathbf{x}(0, \mathbf{p})}{\partial \xi_i} \quad \text{with} \quad \mathbf{s}_i(0) = 0 \quad \text{if } \mathbf{x}(0) \text{ is independent of } \xi_i$$

Sensitivity-Based Control Using Algorithmic Differentiation (1)

- Define the control error

$$J = \sum_{\mu=\nu}^{\nu+N_p} \mathcal{D}\left(\mathbf{y}\left(t_\mu\right) - \mathbf{y}_d\left(t_\mu\right)\right)$$

  between the actual and desired system outputs $\mathbf{y}\left(t\right)$ and $\mathbf{y}_d\left(t\right)$, respectively, to achieve accurate tracking control behavior

- Define the output $\mathbf{y}\left(t\right)$ in terms of the state vector $\mathbf{x}\left(t\right)$ and the control $\mathbf{u}\left(t\right)$ (assumed to be piecewise constant for $t_\nu \leq t < t_{\nu+1}$) according to

$$\mathbf{y}\left(t\right) = \mathbf{g}\left(\mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right)$$

- Compute the differential sensitivity of J using algorithmic differentiation

Sensitivity-Based Control Using Algorithmic Differentiation (2)

- Correct the control input $\mathbf{u}(t_\nu)$ according to

$$\mathbf{u}(t_\nu) = \mathbf{u}(t_{\nu-1}) + \Delta\mathbf{u}_\nu \quad \text{with} \quad \Delta\mathbf{u}_\nu = -\left(\frac{\partial J}{\partial \Delta\mathbf{u}_\nu}\right)^+ \cdot J \ ,$$

where $\mathbf{M}^+ := \left(\mathbf{M}^T\mathbf{M}\right)^{-1}\mathbf{M}^T$ is the left pseudo-inverse of $\mathbf{M}$

- Compute the differential sensitivity of the error measure J

$$\frac{\partial J}{\partial \Delta\mathbf{u}_\nu} = \sum_{\mu=\nu}^{\nu+N_p} \left(\frac{\partial \mathcal{D}\left(\mathbf{g}(\mathbf{x},\mathbf{u}) - \mathbf{y}_d(t_\mu)\right)}{\partial \mathbf{x}} \frac{\partial \mathbf{x}(t_\mu)}{\partial \Delta\mathbf{u}_\nu} + \frac{\partial \mathcal{D}\left(\mathbf{g}(\mathbf{x},\mathbf{u}) - \mathbf{y}_d(t_\mu)\right)}{\partial \Delta\mathbf{u}_\nu} \right)$$

with the property

$$\frac{\partial \mathbf{x}(t_{\nu-1})}{\partial \Delta\mathbf{u}_\nu} = 0$$

- Evaluate $\frac{\partial \mathbf{g}}{\partial \mathbf{x}}$ and $\frac{\partial \mathbf{g}}{\partial \Delta\mathbf{u}_\nu}$ for $\mathbf{x} = \mathbf{x}(t_\mu)$ and $\mathbf{u} = \mathbf{u}(t_{\nu-1}) + \Delta\mathbf{u}_\nu$, $\Delta\mathbf{u}_\nu = 0$

Extensions to Sensitivity-Based Control of Uncertain Systems — Algorithm

Stage 1:

- Allow for uncertainty in parameters and measurements
- Enclose time discretization errors in the computation of the control input

$$\mathbf{u}(t_\nu) = \mathbf{u}(t_{\nu-1}) + \Delta\mathbf{u}_\nu \quad \text{with} \quad \Delta\mathbf{u}_\nu = -\sup\left(\left(\frac{\partial\,[J]}{\partial\Delta\mathbf{u}_\nu}\right)^{+} \cdot [J]\right)$$

Extensions to Sensitivity-Based Control of Uncertain Systems — Algorithm

Stage 1:

- Allow for uncertainty in parameters and measurements
- Enclose time discretization errors in the computation of the control input

$$\mathbf{u}(t_\nu) = \mathbf{u}(t_{\nu-1}) + \Delta\mathbf{u}_\nu \quad \text{with} \quad \Delta\mathbf{u}_\nu = -\sup\left(\left(\frac{\partial [J]}{\partial \Delta\mathbf{u}_\nu}\right)^+ \cdot [J]\right)$$

Stage 2: Check for admissibility of the resulting solution with respect to

- state constraints
- input constraints

Extensions to Sensitivity-Based Control of Uncertain Systems — Algorithm

Stage 1:

- Allow for uncertainty in parameters and measurements
- Enclose time discretization errors in the computation of the control input

$$\mathbf{u}(t_\nu) = \mathbf{u}(t_{\nu-1}) + \Delta\mathbf{u}_\nu \quad \text{with} \quad \Delta\mathbf{u}_\nu = -\sup\left(\left(\frac{\partial\,[J]}{\partial\Delta\mathbf{u}_\nu}\right)^+ \cdot [J]\right)$$

Stage 2: Check for admissibility of the resulting solution with respect to

- state constraints
- input constraints

Stage 3: Adjust the control input if necessary

Extensions to Sensitivity-Based Control of Uncertain Systems — Details

- Quantify the worst-case overshoot over the prediction horizon $t \in \left[t_\nu \; ; \; t_{\nu+\tilde{N}_p}\right]$:

$$\overline{\Delta \mathbf{y}_\nu} := \max_{t \in \left[t_\nu \; ; \; t_{\nu+\tilde{N}_p}\right]} \{0 \; ; \; \sup\left(\left[\mathbf{y}\left(t\right)\right] - \mathbf{y}_d\left(t\right)\right)\}$$

- Evaluate worst-case bounds for the output $\mathbf{y}(t)$, i.e., $\mathbf{y}(t) \in [\mathbf{y}(t)]$ using interval arithmetic techniques

- Adapt the control input according to

$$\Delta \tilde{\mathbf{u}}_\nu = -\sup\left(\left(\frac{\partial\left[\mathbf{y}\right]}{\partial \Delta \tilde{\mathbf{u}}_\nu}\right)^+ \cdot \overline{\Delta \mathbf{y}_\nu}\right)$$

- Re-investigate the admissibility of the control strategy using guaranteed interval enclosures of the output trajectory

Extensions to Sensitivity-Based Control of Uncertain Systems — Example (1)

- Control of a double integrating plant

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ F_d \end{bmatrix} \quad \text{with} \quad m \in [0.9\ ;\ 1.1]\ ,\ \ F_d \in [-0.1\ ;\ 0.1]$$

- Definition of the desired output trajectory

$$y_d(t) = x_{1,d}(t) = 1 - e^{-t}$$

with the initial state

$$\mathbf{x}(0) = \begin{bmatrix} -1 & 0 \end{bmatrix}^T$$

- **Direct computation of a piecewise constant control** with a time-invariant step size $t_{\nu+1} - t_{\nu} = 0.01$ and $N = 200$

Extensions to Sensitivity-Based Control of Uncertain Systems — Example (2)

- Prevent overshooting the desired output trajectory $y_d(t)$ for all $t > 0$ and all possible parameter values $m \in [0.9 \; ; \; 1.1]$
- Use measured state variables $x_{1,m}$ and $x_{2,m}$ during sensitivity computation
- Guaranteed admissibility of the solution in spite of bounded measurement errors

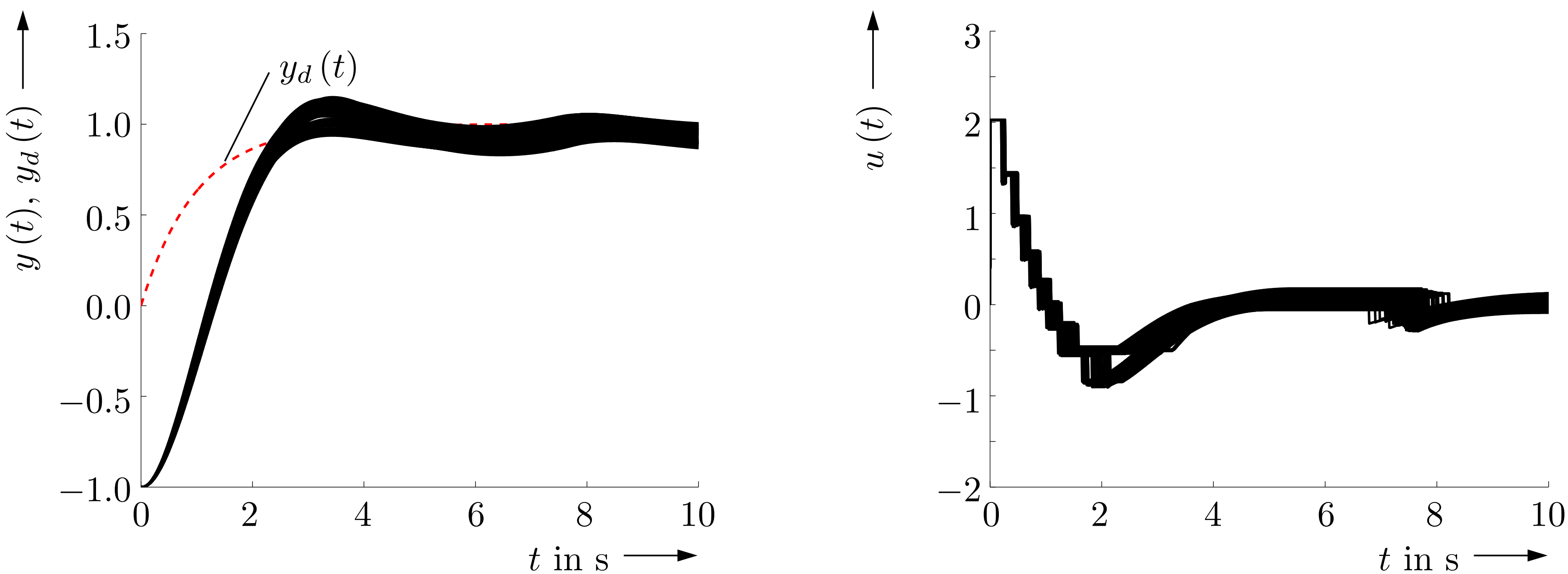
$$x_1(t) \in x_{1,m}(t) + [-0.01 \; ; \; 0.01] \qquad x_2(t) \in x_{2,m}(t) + [-0.01 \; ; \; 0.01]$$

Further algorithmic details:

– A. Rauh, J. Kersten, E. Auer, and H. Aschemann. *Sensitivity Analysis for Reliable Feedforward and Feedback Control of Dynamical Systems with Uncertainties*. In Proc. of 8th Intl. Conference on Structural Dynamics EURODYN 2011, Leuven, Belgium, 2011.
– A. Rauh, J. Kersten, E. Auer, and H. Aschemann. *Sensitivity-Based Feedforward and Feedback Control for Uncertain Systems*. Computing, (2–4):357–367, 2012.

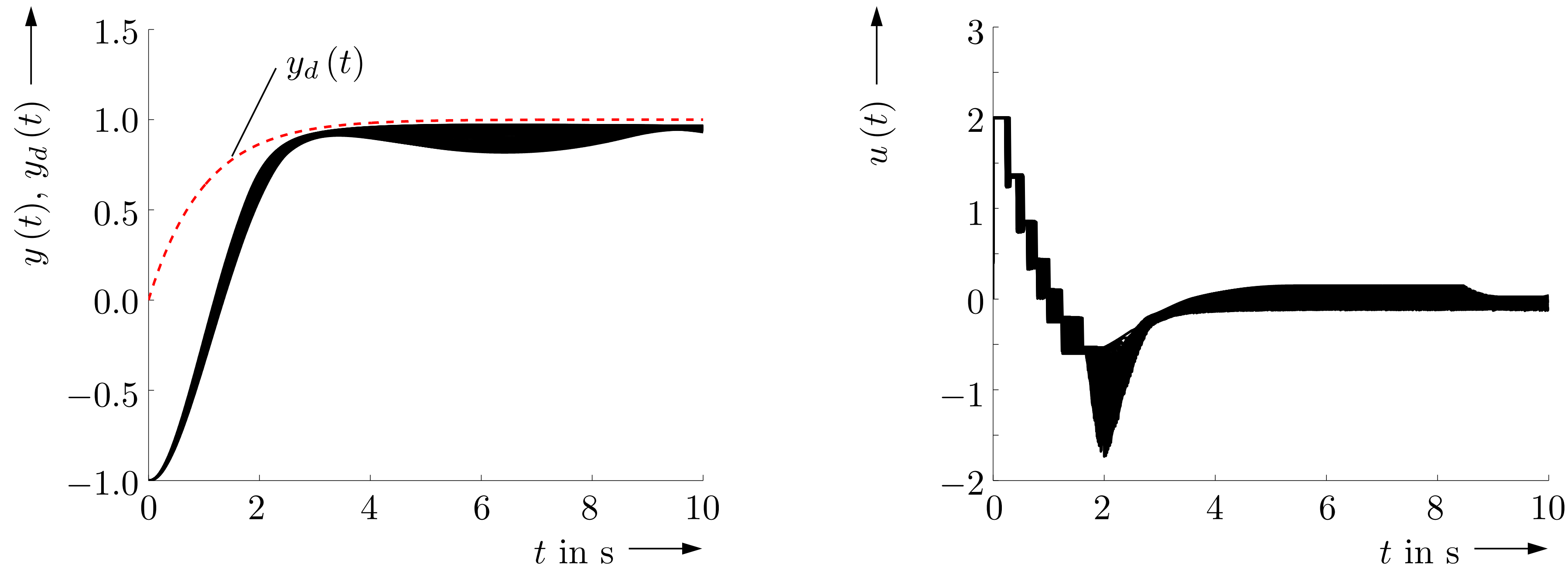
Extensions to Sensitivity-Based Control of Uncertain Systems — Example (3)

Result: Grid-based simulation of sensitivity-based approach without guaranteed overshoot prevention



Extensions to Sensitivity-Based Control of Uncertain Systems — Example (4)

Result: Grid-based validation of sensitivity-based approach with guaranteed overshoot prevention

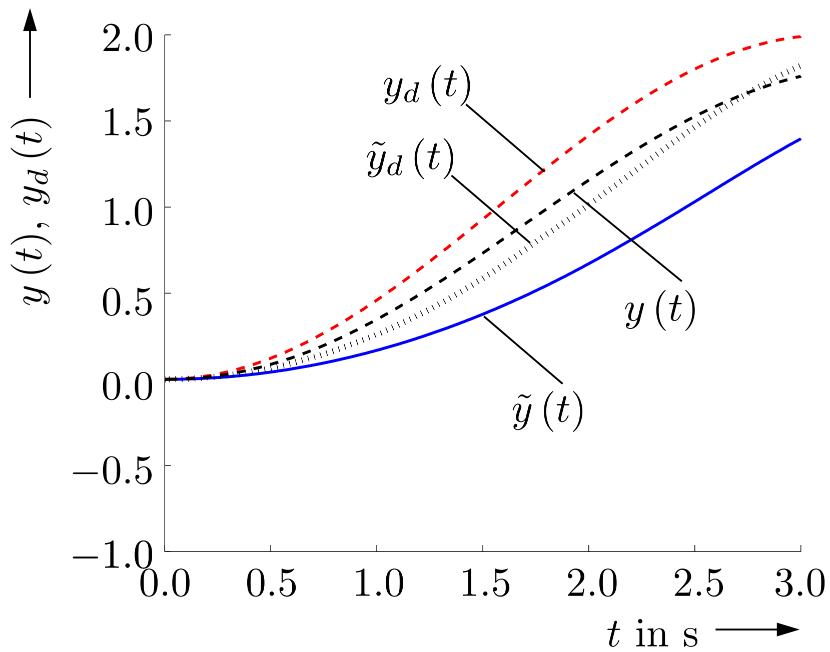


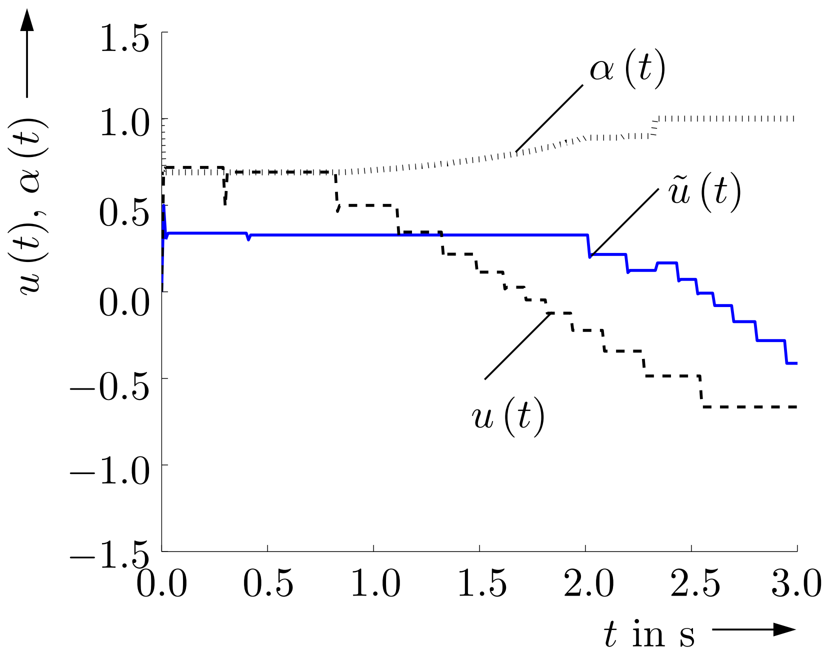
Extension Towards Robust *Path Following* for Uncertain Systems — Example (1)

Procedure: Simultaneous adaptation of the physical control inputs and the desired state trajectory with $u \in [-0.5\ ;\ 0.5]$

Time-scaling by the piecewise constant parameter α_ν as further control input with

$$\tilde{\mathbf{y}}_d\left(t\right) = \mathbf{y}_d\left(\tilde{t}\right) \qquad \text{and} \qquad \tilde{t} = \int_0^t \alpha(\tau) d\tau$$



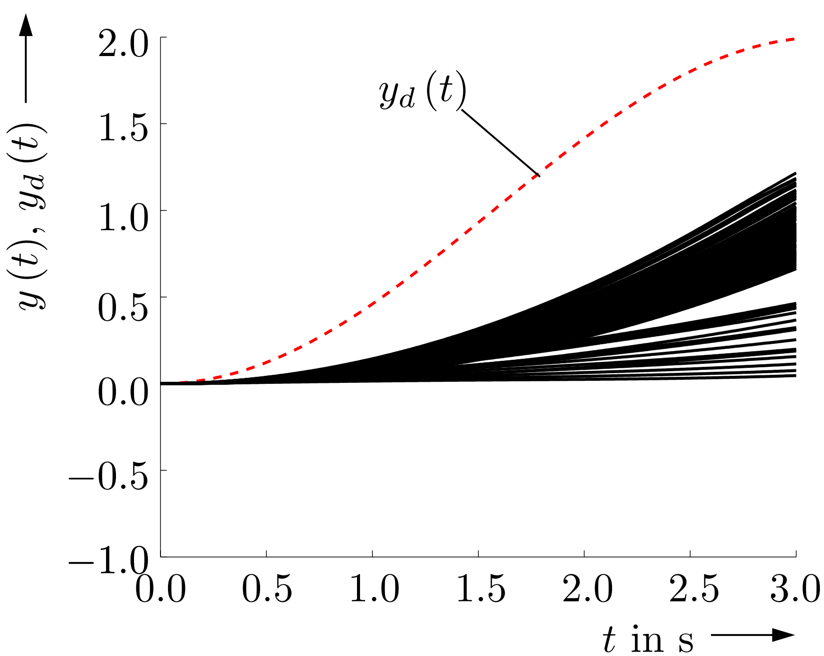


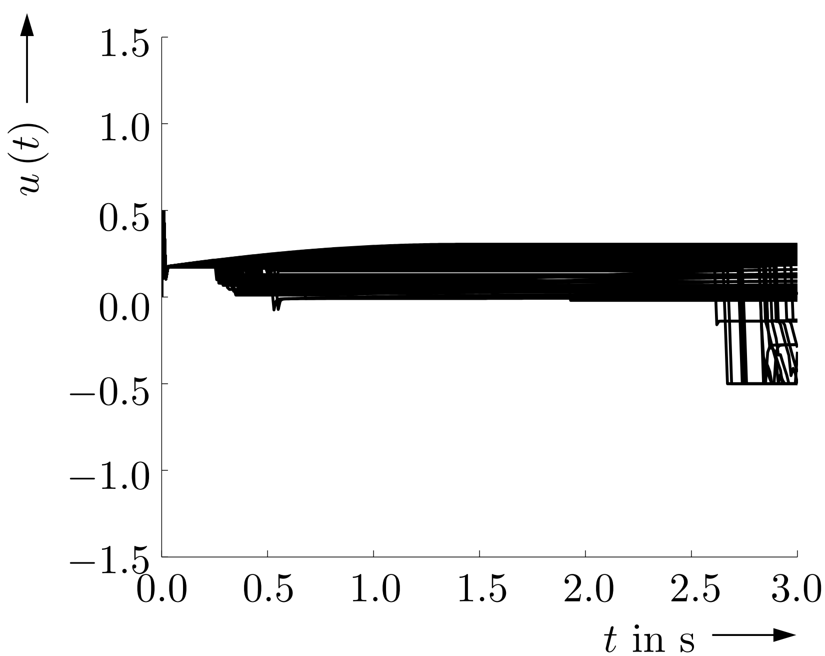
Extension Towards Robust *Path Following* for Uncertain Systems — Example (2)

Procedure: Simultaneous adaptation of the physical control inputs and the desired state trajectory with $u \in [-0.5\ ;\ 0.5]$

Time-scaling by the piecewise constant parameter α_ν as further control input with

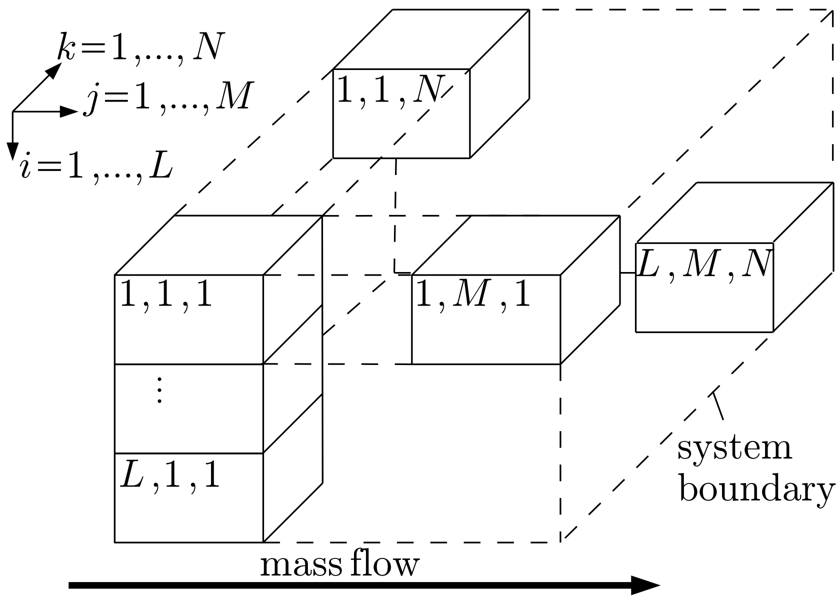
$$\tilde{\mathbf{y}}_d\left(t\right) = \mathbf{y}_d\left(\tilde{t}\right) \qquad \text{and} \qquad \tilde{t} = \int_0^t \alpha(\tau) d\tau$$





Practical Application Scenario: Predictive Temperature for Solid Oxide Fuel Cell Systems (1)

- Control-oriented thermal SOFC model: Semi-discretization into $n_x = L \cdot M \cdot N$ finite volume elements



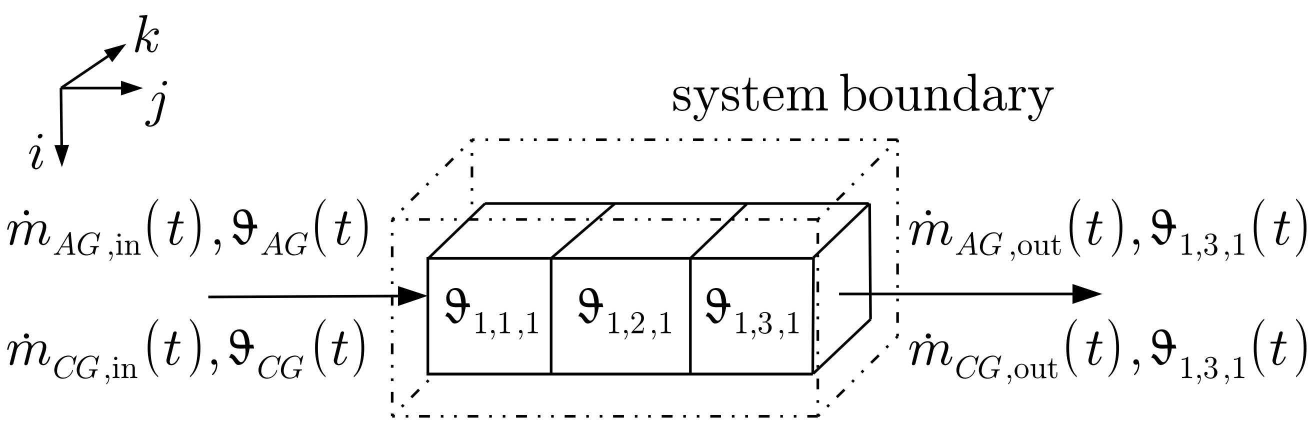
- Introduction of the state vector $\mathbf{x}^T = [\vartheta_{1,1,1}, ..., \vartheta_{L,M,N}] \in \mathbb{R}^{n_x}$ (piecewise homogeneous temperature values)

Practical Application Scenario: Predictive Temperature for Solid Oxide Fuel Cell Systems (2)

- ODE for the local temperature distribution in the stack module

$$
\begin{aligned}
c_{i,j,k} \cdot m_{i,j,k} \cdot \dot{\vartheta}_{i,j,k}(t) &= C_{AG,i,j,k}(\vartheta_{i,j,k}, t) \cdot \Big(\vartheta_{i,j-1,k}(t) - \vartheta_{i,j,k}(t)\Big) \\
&+ C_{CG,i,j,k}(\vartheta_{i,j,k}, t) \cdot \Big(\vartheta_{i,j-1,k}(t) - \vartheta_{i,j,k}(t)\Big) \\
&+ \dot{Q}_{\eta,i,j,k}(t) + \dot{Q}_{R,i,j,k}(t) + P_{El,i,j,k}(t)
\end{aligned}
$$

- Restriction to a system with $n_x = 3$ states (for visualization purposes)



Practical Application Scenario: Predictive Temperature for Solid Oxide Fuel Cell Systems (3)

- Design of a predictive control procedure such that
  - System inputs stay close to the desired set-point

Practical Application Scenario: Predictive Temperature for Solid Oxide Fuel Cell Systems (3)

- Design of a predictive control procedure such that
  - System inputs stay close to the desired set-point
  - Temperature remains close to the desired nominal operating point

Practical Application Scenario: Predictive Temperature for Solid Oxide Fuel Cell Systems (3)

- Design of a predictive control procedure such that
  - System inputs stay close to the desired set-point
  - Temperature remains close to the desired nominal operating point
  - Local violations of the maximum admissible cell temperature are prevented with certainty (in a rigorous formulation)

Practical Application Scenario: Predictive Temperature for Solid Oxide Fuel Cell Systems (3)

- Design of a predictive control procedure such that
  - System inputs stay close to the desired set-point
  - Temperature remains close to the desired nominal operating point
  - Local violations of the maximum admissible cell temperature are prevented with certainty (in a rigorous formulation)
  - Variation rates of the physical system inputs do not violate given bounds (in a weak formulation)

Practical Application Scenario: Predictive Temperature for Solid Oxide Fuel Cell Systems (3)

- Design of a predictive control procedure such that
  - System inputs stay close to the desired set-point
  - Temperature remains close to the desired nominal operating point
  - Local violations of the maximum admissible cell temperature are prevented with certainty (in a rigorous formulation)
  - Variation rates of the physical system inputs do not violate given bounds (in a weak formulation)

- Sensitivity-based manipulation of the **supplied mass flow of cathode gas** as well as the **temperature difference between the preheater and the inlet gas manifold of the SOFC**

Practical Application Scenario: Predictive Temperature for Solid Oxide Fuel Cell Systems (3)

- Design of a predictive control procedure such that
  - System inputs stay close to the desired set-point
  - Temperature remains close to the desired nominal operating point
  - Local violations of the maximum admissible cell temperature are prevented with certainty (in a rigorous formulation)
  - Variation rates of the physical system inputs do not violate given bounds (in a weak formulation)

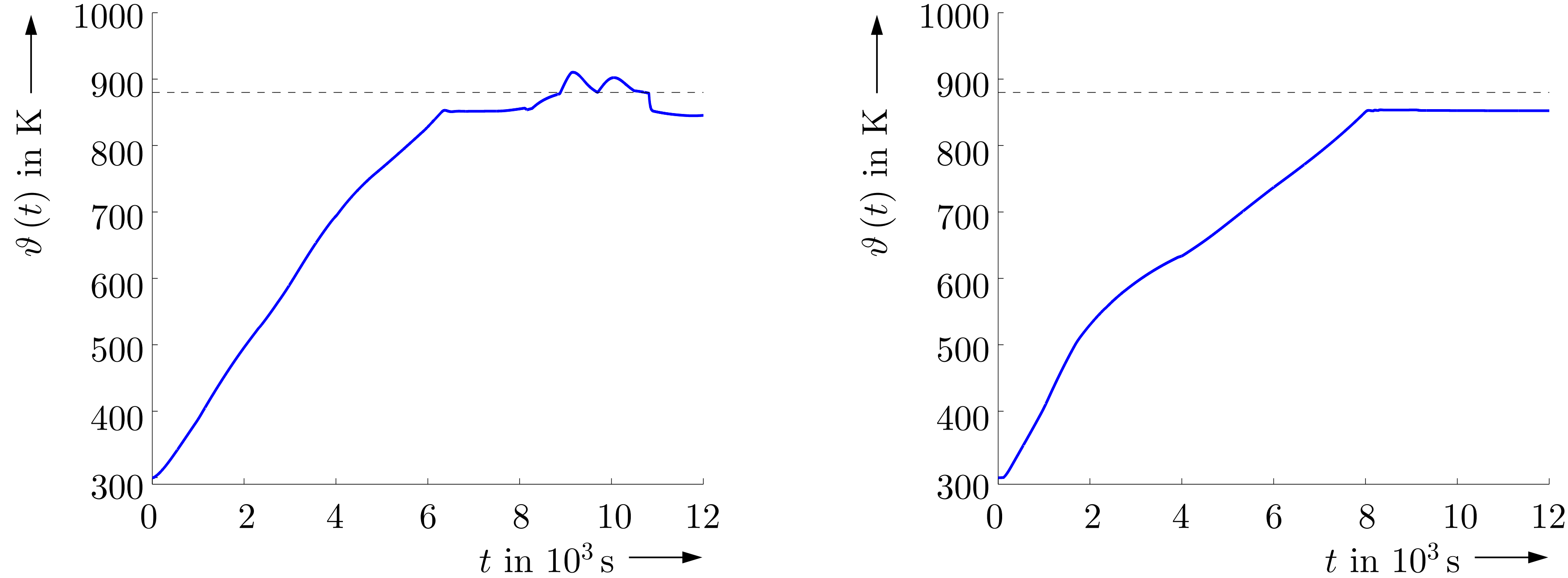
- Sensitivity-based manipulation of the **supplied mass flow of cathode gas** as well as the **temperature difference between the preheater and the inlet gas manifold of the SOFC**

- Alternatively, the enthalpy flow into the SOFC stack module can be computed, which has to be expressed by the physical inputs according to the presentation *Th. Dötschel et al.: Sliding Mode Control for Uncertain Thermal SOFC Models with Physical Actuator Constraints*

Interval-Based Predictive Control (1)

Result: Cell temperature for the scalar system model (desired operating temperature: $850\,\mathrm{K}$, max. admissible temperature $880\,\mathrm{K}$ with varying properties of the anode gas and the electric load)

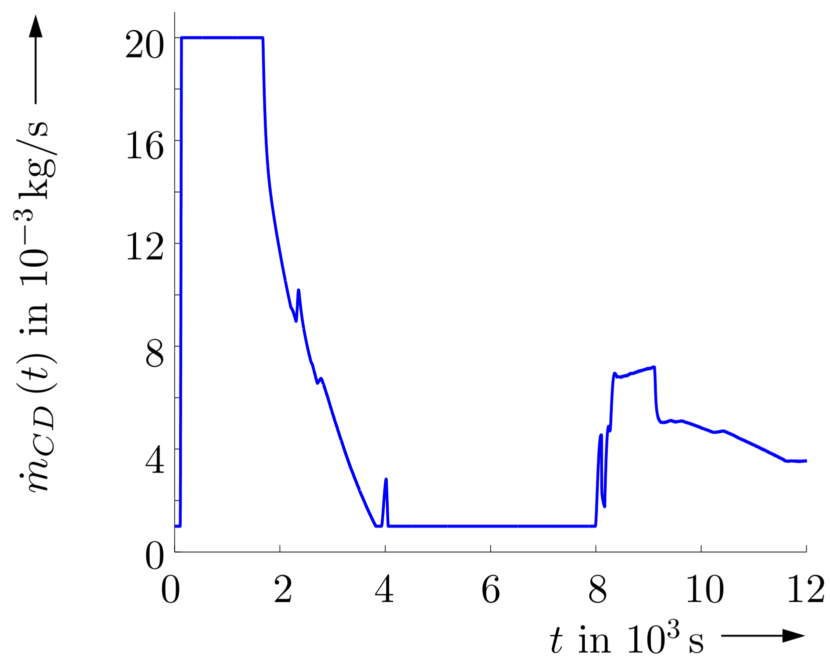
without overshoot prevention *with overshoot prevention*



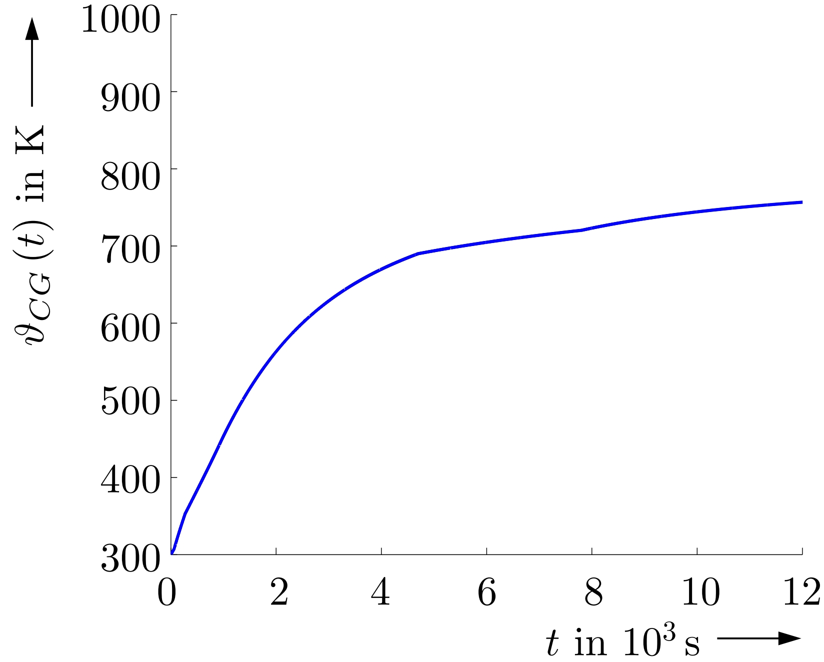
Interval-Based Predictive Control (2)

Result: Cell temperature for the scalar system model (desired operating temperature: $850\,\mathrm{K}$, max. admissible temperature $880\,\mathrm{K}$ with varying properties of the anode gas and the electric load)

mass flow of cathode gas



preheater temperature

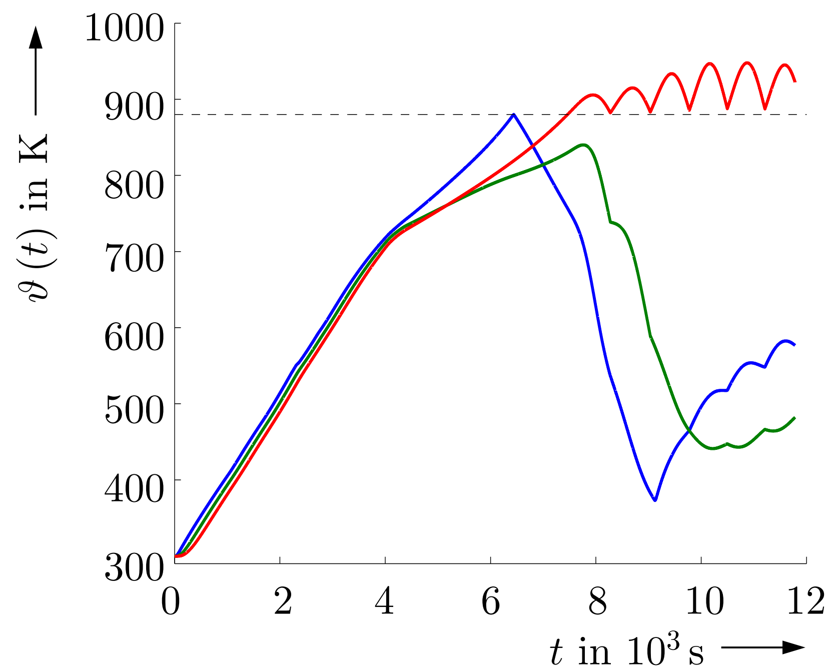


Interval-Based Predictive Control (3)

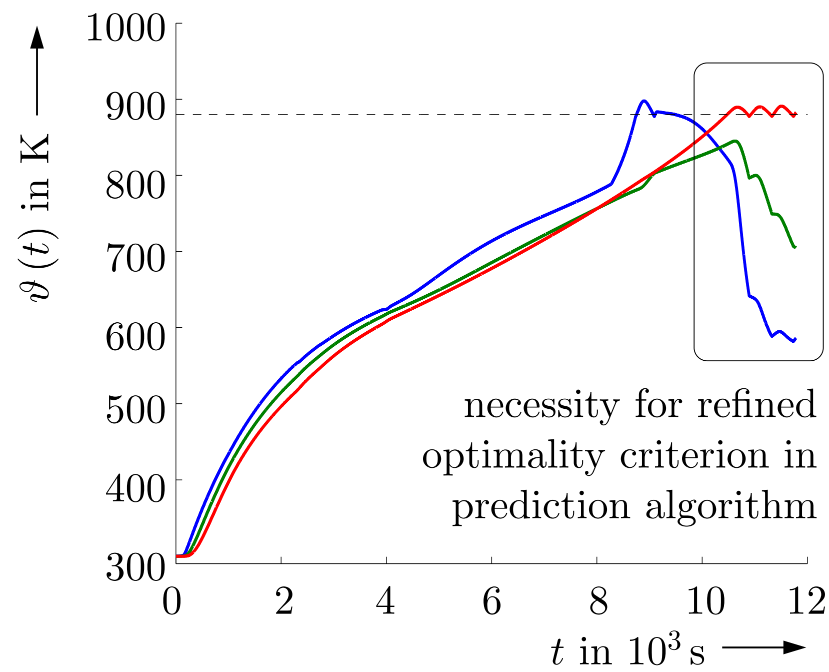
Result: Cell temperature for the system model with $n_x = 3$ states (desired operating temperature: $850\,\mathrm{K}$, max. admissible temperature $880\,\mathrm{K}$ with varying properties of the anode gas and the electric load)

Undesirable behavior after $t = 11,000\,\mathrm{s}$ can be predicted from simulations and avoided by a suitable supervisory control for the remaining system inputs

without overshoot prevention



with overshoot prevention



Derivation of a Physically-Motivated Criterion for the Detection and Reduction of Overestimation (1)

- Prediction of the stack temperatures over the time horizon $t \in \left[t_\nu \; ; \; t_{\nu+N_p}\right]$ with a given number $N_p > 0$ of prediction steps and the constant sampling time $T := t_{\nu+1} - t_\nu$

- Necessity to evaluate the solution to the differential equations specifying the temperature variation rates $\dot{\vartheta}_{i,j,k}(t)$ with uncertain parameters and uncertain initial conditions over the time horizon $t \in \left[t_\nu \; ; \; t_{\nu+N_p}\right]$

$\Longrightarrow$ Overestimation in the state enclosures can make the predictive control procedure inefficient

- Energy-related criterion for the **detection of overestimation**

$$E_\mu := E\left(t_\mu\right) = \sum_{i,j,k} c_{i,j,k} \cdot m_{i,j,k} \cdot \vartheta_{i,j,k}(t_\mu)$$

Derivation of a Physically-Motivated Criterion for the Detection and Reduction of Overestimation (2)

- **Variant 1:** Direct evaluation of

$$E_\mu := E\left(t_\mu\right) = \sum_{i,j,k} c_{i,j,k} \cdot m_{i,j,k} \cdot \vartheta_{i,j,k}(t_\mu)$$

- **Variant 2:** Integral formulation

$$E_\mu = E_\nu + \int\limits_{t_\nu}^{t_\mu} \dot{E}\left(\tau\right) d\tau = E_\nu + \int\limits_{t_\nu}^{t_\mu} \left(\sum_{i,j,k} c_{i,j,k} \cdot m_{i,j,k} \cdot \dot{\vartheta}_{i,j,k}(\tau) \right) d\tau$$

- In the absence of overestimation as well as discretization and rounding errors, both variants yield identical results
- Generally, **variant 2** yields tighter enclosures than **variant 1**

Derivation of a Physically-Motivated Criterion for the Detection and Reduction of Overestimation (3)

- Simplification for state-independent and time-invariant parameters $c_{i,j,k}$ and $m_{i,j,k}$ which are identical for all finite volume elements
- Modified formulation
  - **Variant 1:** Direct evaluation of

$$E_\mu := E\left(t_\mu\right) = \sum_{i,j,k} \vartheta_{i,j,k}(t_\mu)$$

  - **Variant 2:** Integral formulation

$$E_\mu = E_\nu + \int_{t_\nu}^{t_\mu} \dot{E}\left(\tau\right) d\tau = E_\nu + \int_{t_\nu}^{t_\mu} \left(\sum_{i,j,k} \dot{\vartheta}_{i,j,k}(\tau) \right) d\tau$$

- Determine the offset $E_\nu \in [E_\nu]$ on the basis of measured temperatures (including measurement tolerances and estimation errors)

Derivation of a Physically-Motivated Criterion for the Detection and Reduction of Overestimation (4)

- Simplification for state-independent and time-invariant parameters $c_{i,j,k}$ and $m_{i,j,k}$ which are identical for all finite volume elements
- Modified formulation
  - **Variant 1:** Direct evaluation of

$$E_\mu := E\left(t_\mu\right) = \sum_{i,j,k} \vartheta_{i,j,k}(t_\mu)$$

  - **Variant 2:** Integral formulation

$$E_\mu = E_\nu + \int_{t_\nu}^{t_\mu} \dot{E}\left(\tau\right) d\tau = E_\nu + \int_{t_\nu}^{t_\mu} \left(\sum_{i,j,k} \dot{\vartheta}_{i,j,k}(\tau) \right) d\tau$$

- Reduced overestimation on **variant 2** since the heat flow over boundaries between neighboring finite volume elements cancels out exactly (energy conservation: first law of thermodynamics!)

Discrete-Time Formulation of the Predictive Control Algorithm (1)

- Determine state enclosure for $t = t_\nu$: $\vartheta_{i,j,k}(t_\nu) \in [\vartheta_{i,j,k}(t_\nu)]$
- Discrete-time evaluation of the state equations over the complete prediction horizon $\left[t_\nu \; ; \; t_{\nu+N_p}\right]$, $\mu > \nu$

$$\vartheta_{i,j,k}(t_\mu) \in [\vartheta_{i,j,k}(t_{\mu-1})] + T \cdot \left[\dot{\vartheta}_{i,j,k}(t_{\mu-1})\right] \quad \text{with} \quad \mathbf{u} = \mathbf{u}(t_{\nu-1})$$

- Simultaneous evaluation of the performance criterion
- Evaluation of the corresponding sensitivities by means of algorithmic differentiation

Discrete-Time Formulation of the Predictive Control Algorithm (2)

- Simultaneous evaluation of the energy-related overestimation criterion
  - **Variant 1:** Direct evaluation for discrete-time state intervals

$$E_\mu \in \sum_{i,j,k} [\vartheta_{i,j,k}(t_\mu)]$$

  - **Variant 2:** Integral formulation

$$E_\mu \in \left[\tilde{E}_\mu\right] := [E_\nu] + \sum_{\mu'=\nu}^{\mu} \left(\sum_{i,j,k} \left[\dot{\vartheta}_{i,j,k}(t'_\mu)\right] \right)$$

- As in the continuous-time case, **variant 2** yields tighter enclosures do to reduction of the wrapping effect (elimination of internal heat flow in the SOFC)

Discrete-Time Formulation of the Predictive Control Algorithm (3)

- Reduction of the conservativeness with respect to the maximum predicted overshoot for $t \in \left[t_\nu \,;\, t_{\nu+N_p}\right]$ at $t = t_{\mu^*}$ by the following consistency test
  - Subdivide $[\vartheta_{i,j,k}(t_{\mu^*})]$ into subintervals $\left[\vartheta'_{i,j,k}(t_{\mu^*})\right]$ along the longest edge
  - Evaluate

$$E'_\mu \in [E'_\mu] = \sum_{i,j,k} [\vartheta'_{i,j,k}(t_\mu)]$$

    for all subintervals of the predicted state enclosure $[\vartheta_{i,j,k}(t_\mu)]$

Discrete-Time Formulation of the Predictive Control Algorithm (3)

- Reduction of the conservativeness with respect to the maximum predicted overshoot for $t \in \left[t_\nu \ ; \ t_{\nu+N_p}\right]$ at $t = t_{\mu^*}$ by the following consistency test
  - Subdivide $[\vartheta_{i,j,k}(t_{\mu^*})]$ into subintervals $\left[\vartheta'_{i,j,k}(t_{\mu^*})\right]$ along the longest edge
  - Evaluate
$$E'_\mu \in [E'_\mu] = \sum_{i,j,k} [\vartheta'_{i,j,k}(t_\mu)]$$
    for all subintervals of the predicted state enclosure $[\vartheta_{i,j,k}(t_\mu)]$

- Classification of the resulting subintervals
  - Guaranteed caused by overestimation if $[E'_\mu] \cap \left[\tilde{E}_\mu\right] = \emptyset$
  - Undecided for $[E'_\mu] \cap \left[\tilde{E}_\mu\right] \neq \emptyset$ and $[E'_\mu] \not\subseteq \left[\tilde{E}_\mu\right]$
  - Consistent for $[E'_\mu] \subseteq \left[\tilde{E}_\mu\right]$

Discrete-Time Formulation of the Predictive Control Algorithm (3)

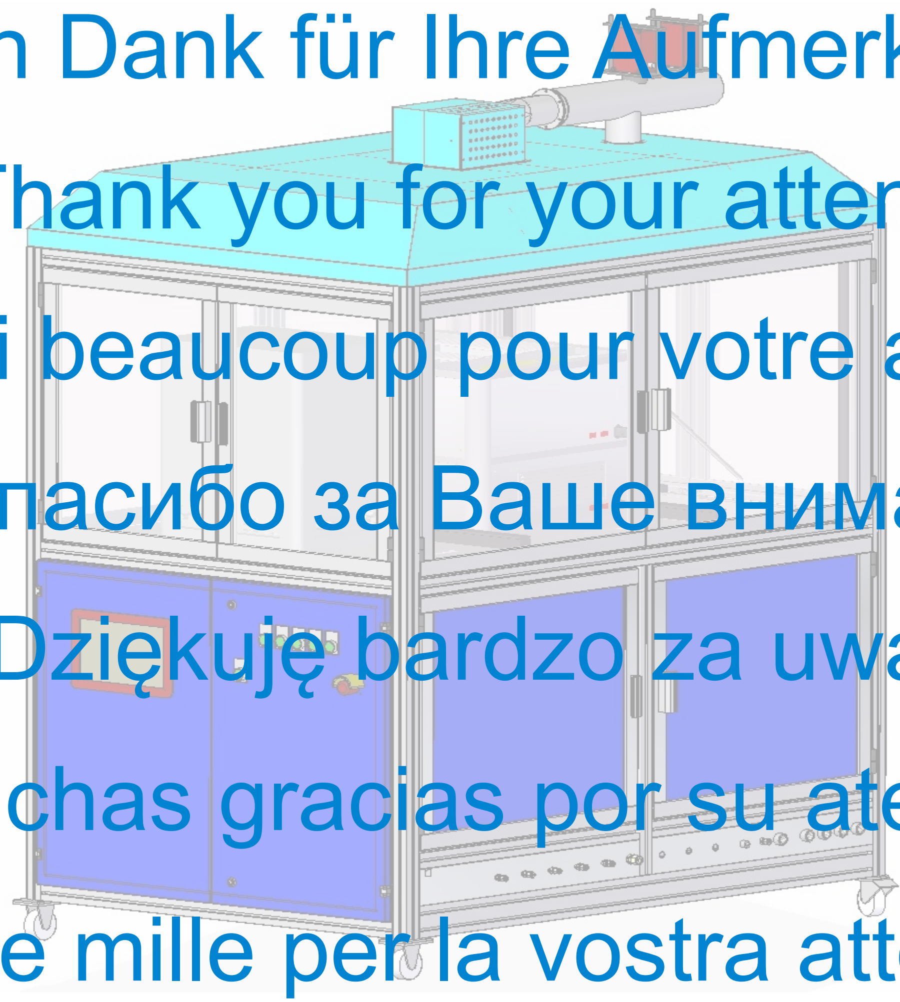
- Reduction of the conservativeness with respect to the maximum predicted overshoot for $t \in \left[t_\nu \; ; \; t_{\nu+N_p}\right]$ at $t = t_{\mu^*}$ by the following consistency test
  - Subdivide $[\vartheta_{i,j,k}(t_{\mu^*})]$ into subintervals $\left[\vartheta'_{i,j,k}(t_{\mu^*})\right]$ along the longest edge
  - Evaluate
  $$E'_\mu \in [E'_\mu] = \sum_{i,j,k} [\vartheta'_{i,j,k}(t_\mu)]$$
  for all subintervals of the predicted state enclosure $[\vartheta_{i,j,k}(t_\mu)]$
- Classification of the resulting subintervals
  - Guaranteed caused by overestimation if $[E'_\mu] \cap \left[\tilde{E}_\mu\right] = \emptyset$
  - Undecided for $[E'_\mu] \cap \left[\tilde{E}_\mu\right] \neq \emptyset$ and $[E'_\mu] \not\subseteq \left[\tilde{E}_\mu\right]$
  - Consistent for $[E'_\mu] \subseteq \left[\tilde{E}_\mu\right]$
- Re-evaluate $[J]$ for the reduced predicted overshoot
  $\Longrightarrow$ Perform the sensitivity-based control update as for the illustrative example

Conclusions and Outlook on Future Work

- Framework for sensitivity-based open-loop and closed-loop control with real-life applications
- Extension of sensitivity-based control to systems with interval uncertainties
  $\Longrightarrow$ Guarantee the compliance with state and control constraints
- Development of a general framework for interval arithmetic, sensitivity-based model-predictive control
  $\Longrightarrow$ Problem-dependent definition of corresponding cost functions

Conclusions and Outlook on Future Work

- Framework for sensitivity-based open-loop and closed-loop control with real-life applications
- Extension of sensitivity-based control to systems with interval uncertainties
  $\Longrightarrow$ Guarantee the compliance with state and control constraints
- Development of a general framework for interval arithmetic, sensitivity-based model-predictive control
  $\Longrightarrow$ Problem-dependent definition of corresponding cost functions
- Extension of sensitivity-based control to state and disturbance estimation (duality of control and observer synthesis)
- Verification of (asymptotic) stability
- Gain scheduling for sliding mode control with interval uncertainties



Vielen Dank für Ihre Aufmerksamkeit!

Thank you for your attention!

Merci beaucoup pour votre attention!

Спасибо за Ваше внимание!

Dziękuję bardzo za uwagę!

¡muchas gracias por su atención!

Grazie mille per la vostra attenzione!