Slovak University of Technology in Bratislava
Institute of Information Engineering, Automation, and Mathematics

Application of Machine Learning in Accelerating MPC for Chemical Processes

Martin Klaučo, Patrik Valábek

martin.klauco@stuba.sk

Acknowledgments: VEGA 1/0239/24, VEGA 1/0490/23, APVV-21-0019, APVV-20-0261, FrontSeat (HEU 101079342)

Speakers

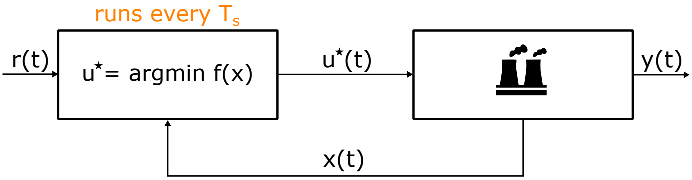
- **Lecture & overview**: Martin Klaučo, `martin.klauco@stuba.sk`
- **Hands-on exercises**: Patrik Valábek, `patrik.valabek@stuba.sk`

Notation

- u – control input or manipulated variable
- $u^\star$ – optimal control input
- $\tilde{u}$ – approximated control input
- $x, \theta, \hat{x}$ – state variables, and if not measured, then estimated
- y – output variable
- r – reference

Why Accelerating the MPC?

Comparison - MPC



How to Accelerate the MPC?

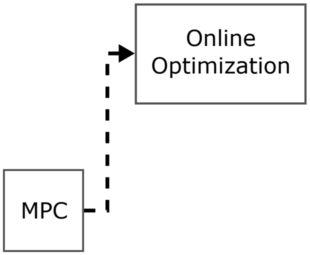
Holy Grail of Control

- Multiple-input Multiple-output Controllers
- Optimal control action (s.t. performance, contraints, etc.)
- Fast evaluation
- Low memory footprint

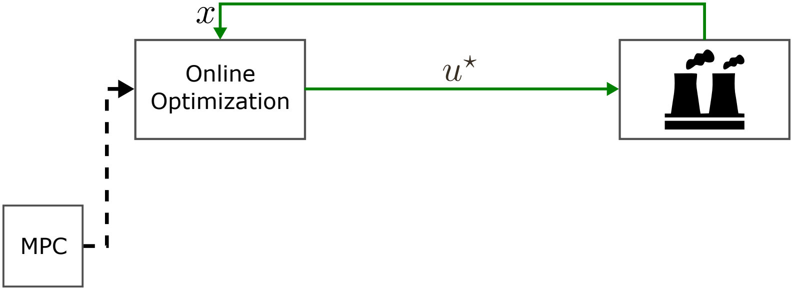
- $u^\star = f(\theta)$, where $f(\theta)$ is explicit function

Pros/Cons of MPC-based Control

Pros/Cons of MPC-based Control



Pros/Cons of MPC-based Control

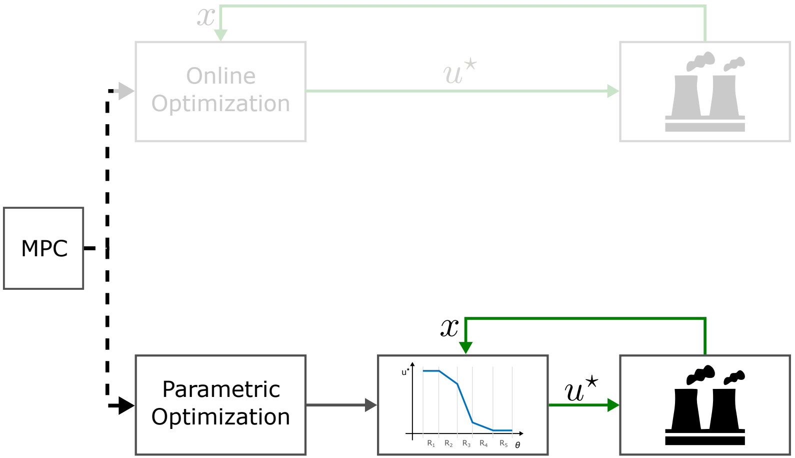


Arbitrary problem size

Tunability

Online optimization

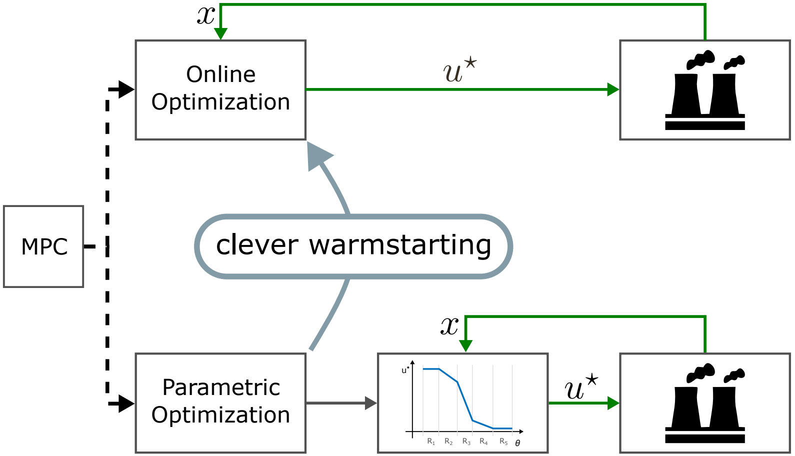
Pros/Cons of MPC-based Control



Arbitrary problem size
Tunability
Online optimization

Fast execution
Tunability
Gigabytes
Small problems

Sidenote – Pros/Cons of MPC-based Control



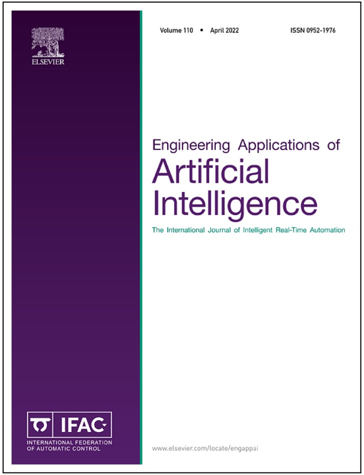
Arbitrary problem size
Tunability
Online optimization

Fast execution
Tunability
Gigabytes
Small problems

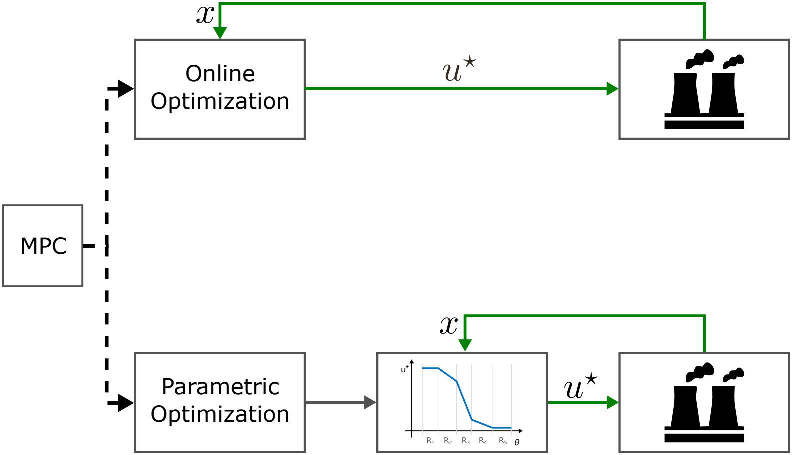
Sidenote – Accelerating MPC with ML-based warmstarting

- Directly providing constraints to the ASM solver.
- Drastic reduction of the number of iterations.
- Guaranteed convergence.
- Increased memory footprint.

- M. Klaučo – M. Kalúz – M. Kvasnica: Machine learning-based warm starting of active set methods in embedded model predictive control. Engineering Applications of Artificial Intelligence, vol. 77, page. 1-8, 2019.



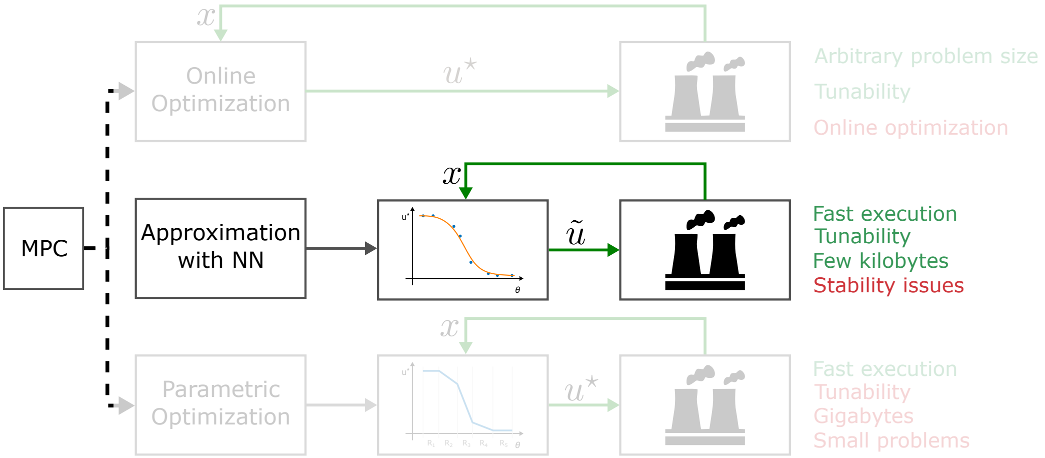
Pros/Cons of MPC-based Control



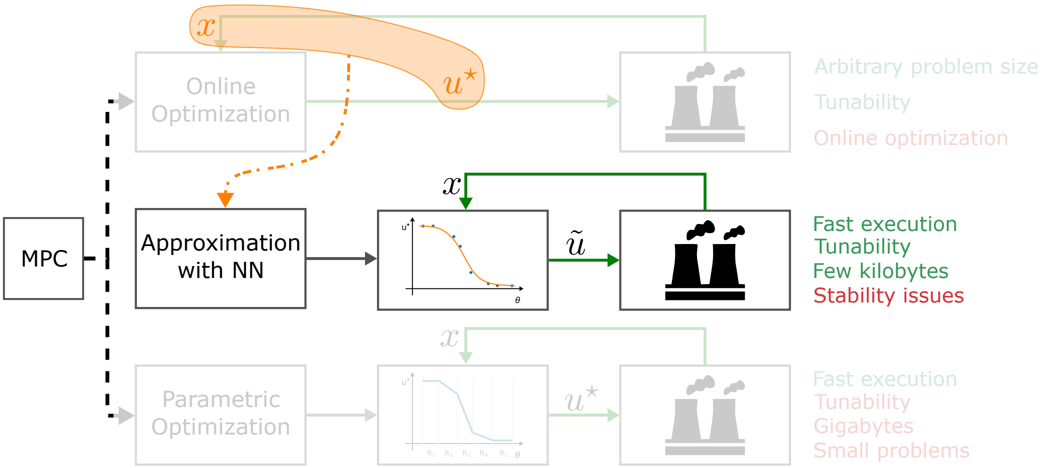
Arbitrary problem size
Tunability
Online optimization

Fast execution
Tunability
Gigabytes
Small problems

Pros/Cons of MPC-based Control



Pros/Cons of MPC-based Control

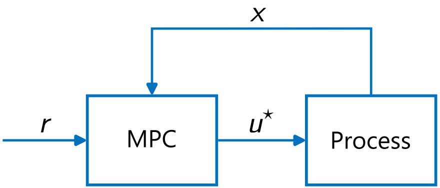


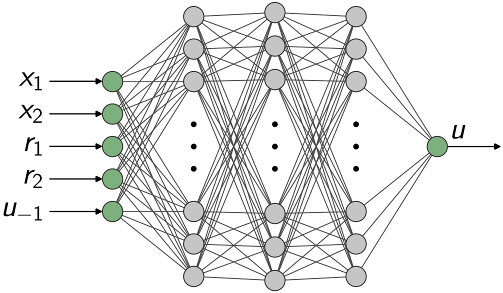
Training of Neural Network Control Law

Repeatedly solve MPC for
large pool of initial conditions

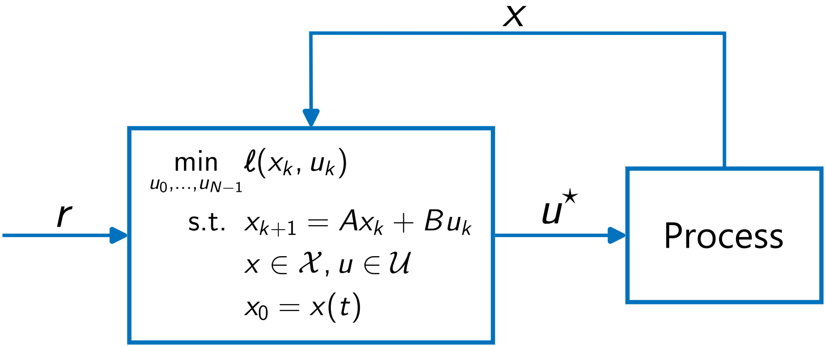
Train neural network
in offline mode

$$\min \left| u_{\mathrm{NN}} - u^\star \right|$$

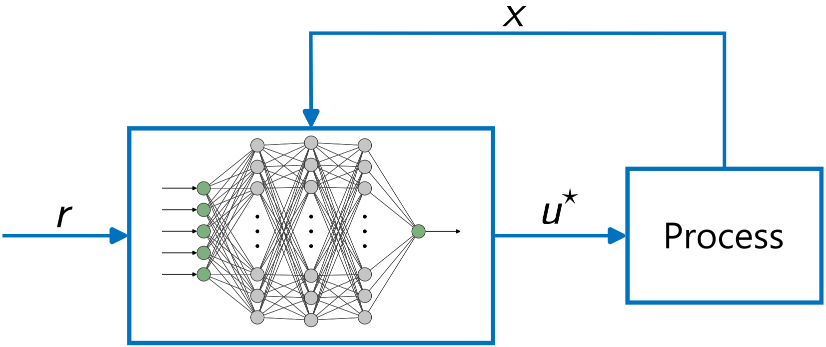




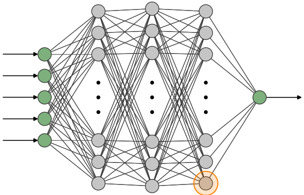
Closed-loop Arrangement with Neural Network

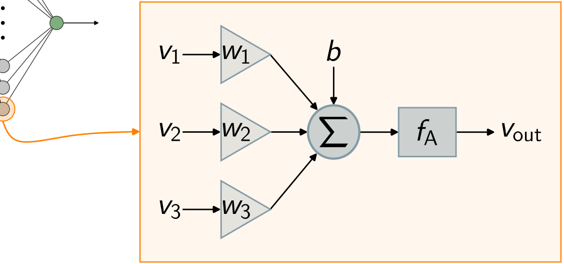


Closed-loop Arrangement with Neural Network



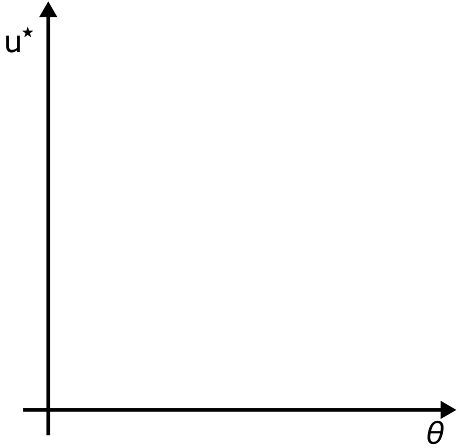
Single Neuron



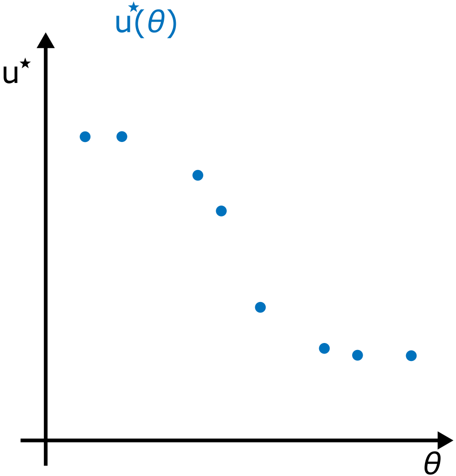


Is it a feasible approach? Can we do that?

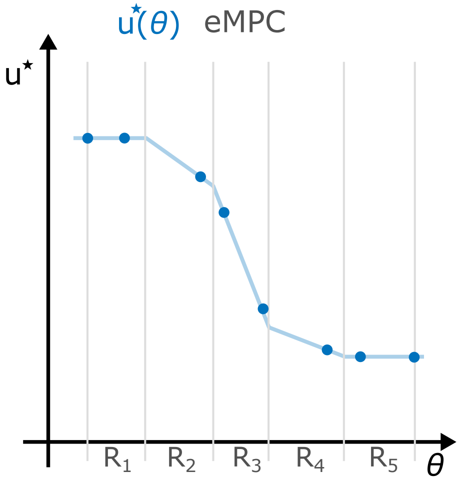
Explicit Controllers



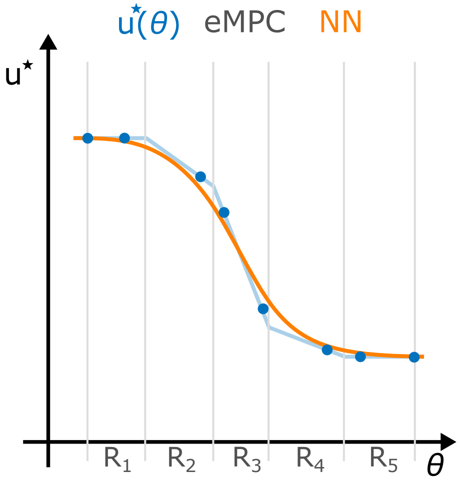
Explicit Controllers



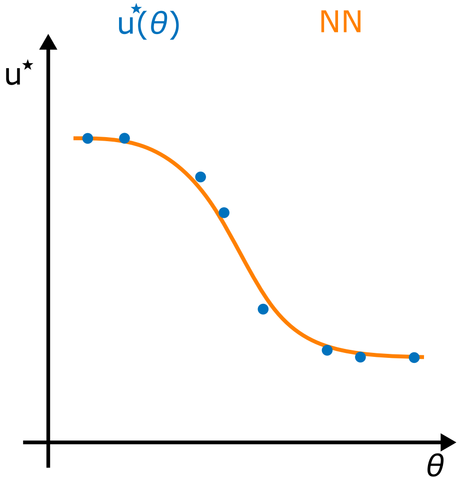
Explicit Controllers



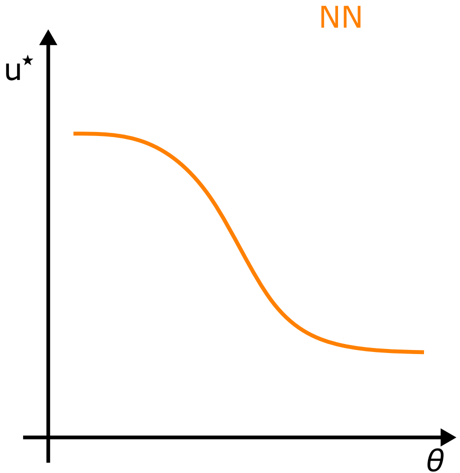
Explicit Controllers



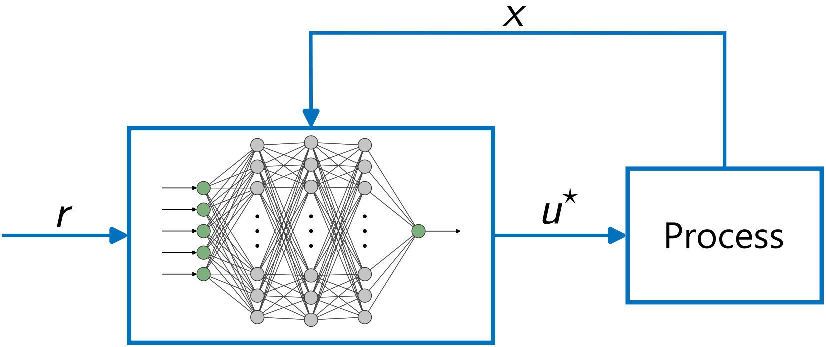
Explicit Controllers



Explicit Controllers



Closed-loop Arrangement with Neural Network

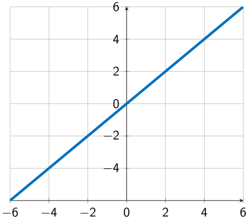


Activation Functions

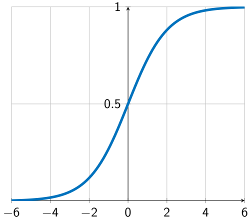
Type	Expression
Linear function	$f_{\mathsf{A}}(\gamma) = \gamma$
Sigmoid function	$f_{\mathsf{A}}(\gamma) = \dfrac{1}{1 + \mathrm{e}^{-\gamma}}$
Hyperbolic Tangent	$f_{\mathsf{A}}(\gamma) = \dfrac{e^{2\gamma-1}}{e^{2\gamma+1}}$
Rectified Linear Unit	$f_{\mathsf{A}}(\gamma) = \max(0, \gamma)$

$$\gamma = \sum_{i=1}^{n_{\mathrm{v}}} w_i \cdot v_i + b$$

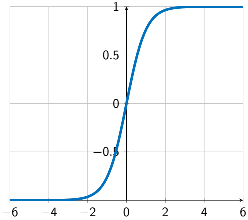
Activation Functions



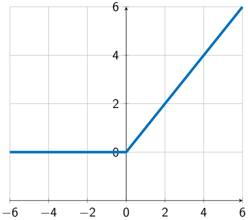
(a) Linear function



(b) Sigmoid function



(c) Hyperbolic tangent function



(d) ReLU function

Non-linear MPC

$$
\begin{aligned}
\min \quad & \sum_{k=0}^{N-1} (y_k - y_{\mathrm{ref}})^\intercal Q_{\mathrm{y}} (y_k - y_{\mathrm{ref}}) + \sum_{k=0}^{N-1} \Delta u_k^\intercal Q_{\mathrm{du}} \Delta u_k \\
\text{s.t.} \quad & x_{k+1} = f(x_k, u_k),\ k \in \mathbb{N}_0^{N-1} \\
& y_k = g(x_k, u_k),\ k \in \mathbb{N}_0^{N-1} \\
& u_k \in \mathcal{U},\ \Delta u_k \in \mathcal{U}_\Delta,\ x_k \in \mathcal{X},\ y_k \in \mathcal{Y},\ k \in \mathbb{N}_0^{N-1} \\
& \Delta u_k = u_k - u_{k-1},\ k \in \mathbb{N}_0^{N-1} \\
& x_0 = x(t),\ y_{\mathrm{ref}} = r(t),\ u_{-1} = u(t - T_{\mathrm{s}})
\end{aligned}
$$

Preparation of NN-based Control Law

set of initial conditions

$$
\begin{aligned}
x_0 &\in \mathcal{X} \\
r &\in \mathscr{R} \\
u_{-1} &\in \mathcal{U}
\end{aligned}
$$

Preparation of NN-based Control Law

set of initial conditions

$$x_0 \in \mathcal{X}$$
$$r \in \mathcal{R}$$
$$u_{-1} \in \mathcal{U}$$

optimization

set of optimal trajectories

x_0	r	u_{-1}	$u^\star$
•	•	•	•
•	•	•	•
•	•	•	•
•	•	•	•
•	•	•	•
•	•	•	•
•	•	•	•
•	•	•	•

Preparation of NN-based Control Law

set of initial conditions

$$x_0 \in \mathcal{X}$$
$$r \in \mathcal{R}$$
$$u_{-1} \in \mathcal{U}$$

optimization

set of optimal trajectories

x_0	r	u_{-1}	$u^\star$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

features

targets

Preparation of NN-based Control Law

set of initial conditions

$$x_0 \in \mathcal{X}$$
$$r \in \mathcal{R}$$
$$u_{-1} \in \mathcal{U}$$

optimization

set of optimal trajectories

x_0	r	u_{-1}	$u^\star$
⋮	⋮	⋮	⋮

features

targets

machine learning

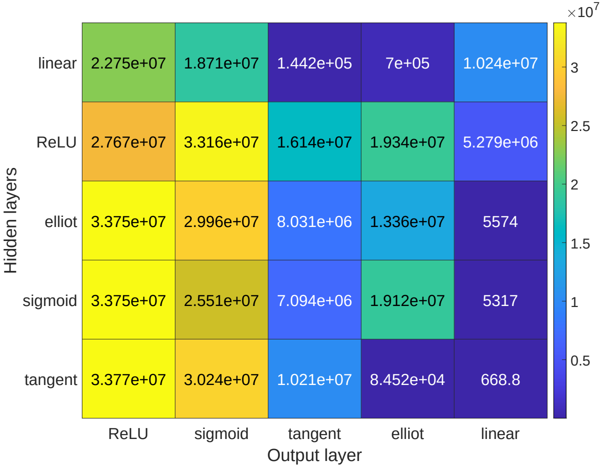
generated neural net

$$\tilde{u} = f(\theta)$$

Pros/Cons of the Approximation with Neural Networks

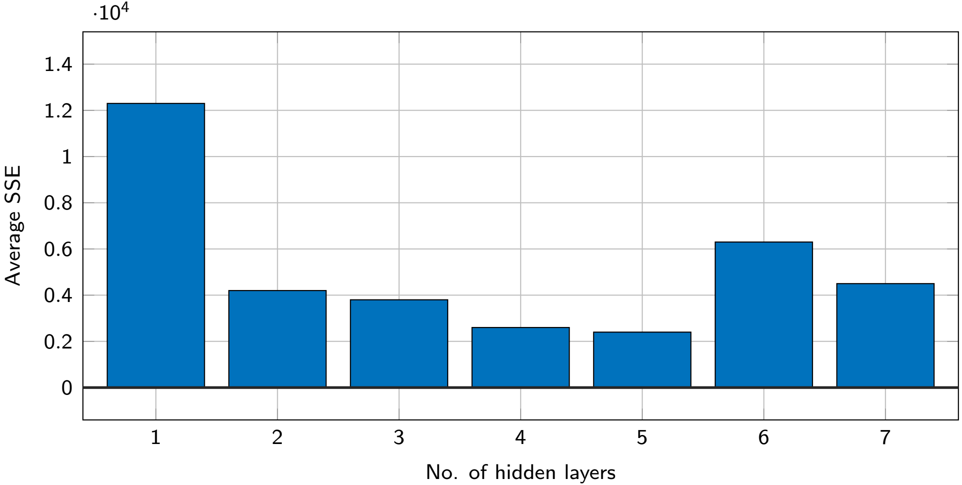
- Replacing the optimization
- Reducing the evaluation to a single function call
- LTI-based MPC & non-linear MPC can be approximated
- Data generation – NN can interpolate not extrapolate
- Structure of the NN is important and only some rules of thumb are available

Choice of the Structure of NN-based Controller #1



K. Kiš – M. Klaučo – A. Mészáros: *Neural Network Controllers in Chemical Technologies*. In 2020 IEEE 15th International Conference of System of Systems Engineering, IEEE, pp. 397–402, 2020.

Choice of the Structure of NN-based Controller #1



K. Kiš – M. Klaučo – A. Mészáros: *Neural Network Controllers in Chemical Technologies*. In 2020 IEEE 15th International Conference of System of Systems Engineering, IEEE, pp. 397–402, 2020.

Training of the Neural Network – NLP

$$\min_{w_i, bi} \sum_{j=1}^{M} |u_{\mathrm{NN},j} - u_j^\star|$$
$$\text{s.t. } u_{\mathrm{NN},j} = F(\theta_j), \quad \forall j \in \mathbb{N}_1^M$$

where

$$F(\theta) = f_{\mathrm{A},1}\Big(f_{\mathrm{A},2}(\ldots), \ldots\Big)$$

objective function alternatives

$$\ell(\theta, u^\star) = \frac{1}{n}\left(u_{\mathrm{NN},j} - u_j^\star\right)^2$$

solved usually by Stochastic Gradient Descent Method

Further Reading

- K. Kiš - P. Bakaráč - M. Klaučo: Nearly Optimal Tunable MPC Strategies on Embedded Platforms. In 18th IFAC Workshop on Control Applications of Optimization, IFAC-PapersOnline, pp. 326-331, 2022, LINK
- Y. Lohr - M. Klaučo - M. Fikar - M. Mönnigmann: Machine Learning Assisted Solutions of Mixed Integer MPC on Embedded Platforms. IFAC World Congress 2020, LINK
- Sergio Lucia - Benjamin Karg: A deep learning-based approach to robust nonlinear model predictive control, 6th IFAC Conference on Nonlinear Model Predictive Control NMPC 2018, LINK

Software to Install

- Matlab
- tbxManager https://www.tbxmanager.com/
- YALMIP `tbxmanager install yalmip`
- (optional) MPT3 `tbxmanager install mpt mptdoc cddmex fourier glpkmex hysdel lcp yalmip sedumi espresso`
- Statistics and Machine Learning Toolbox, Deep Learning Toolbox, Control Toolbox, Optimization Toolbox

Hands-on exercise

- Concept of neural networks
- Role of neural networks as controllers
- Data generation and NN training in MATLAB
- Deployment of NN-based control laws in MATLAB

Enjoy, learn, and keep in touch

martin.klauco@stuba.sk & patrik.valabek@stuba.sk

www.uiam.sk