25 - 28 June, 2024
Stockholm | Sweden

Towards neural networks for modeling and control: stability, safety verification, adaptation

Riccardo Scattolini

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano

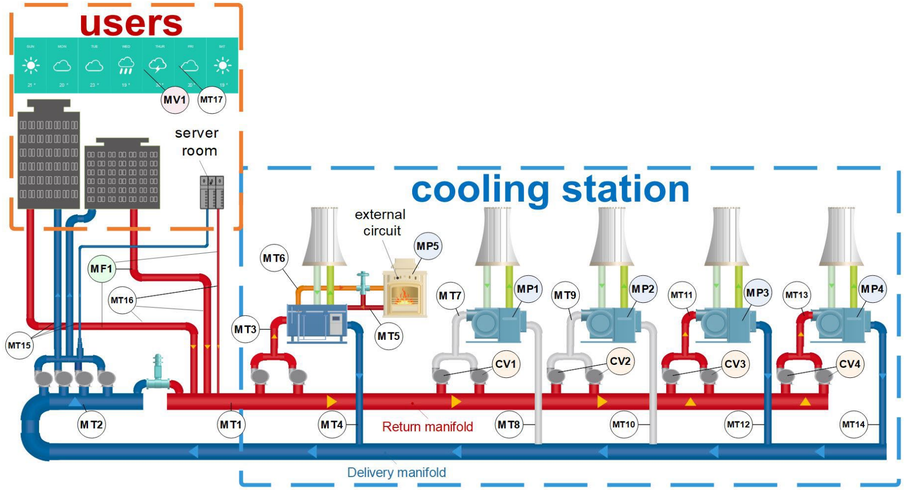
Outline of the talk

- **Motivations and goals for NN-based control**
- Recurrent Neural Networks
- Stability, verification, generalization
- Physics informed RNN
- Lifelong learning and adaptation
- Conclusions

Motivations

Modeling and control of the cooling system of a large commercial center

Five buildings, hundreds of meters pipes



Plant topology almost unknown (cooling circuit), no knowledge on the position of the pipes ⟶ almost impossible to derive and ***TUNE*** a physical model

Goals

- To derive structural properties (***training procedures, stability, verification, interpretability, adaptation, …***) to make RNN a standard tool for the design of controllers for complex systems

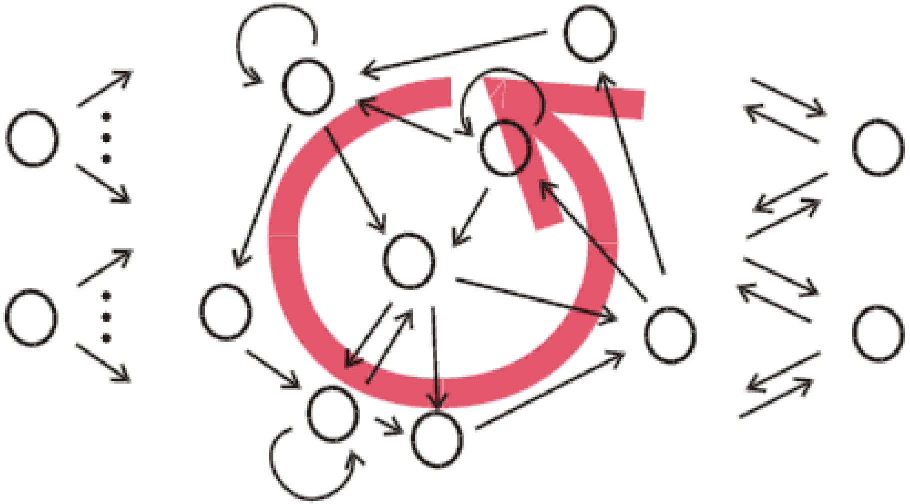
- To use RNN structures ***already widely used*** in many engineering fields (and not only) for regression and classification:

  - Echo State networks
  - NNARX models
  - Long Short Term Memory networks (LSTM)
  - Gated Recurrent Units (GRU) ⟵

Outline of the talk

- Motivations and goals for NN-based control
- **Recurrent Neural Networks**
- Stability, verification, generalization
- Physics informed RNN
- Lifelong learning and adaptation
- Conclusions

Recurrent Neural Networks



$$\begin{cases} x_i(k+1) = f_i(x_i(k), u_i(k), \Phi_i) \\ y_i(k) = g_i(x_i(k), u_i(k), \Phi_i) \end{cases}$$

Activation functions

$$\sigma_g(x) = \frac{1}{1+e^{-x}}$$

$$\sigma_c(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Cyclic paths, loop of information

Specifically developed for dynamic systems

NN for control design

NN as models of the plant (indirect approach)

NN as models of the model uncertainty

NN as approximators of an off-line computed control law (explicit MPC)

NN as models of the controller directly synthesized from data (direct approach)

NN as tools to approximate dynamic programming (reinforcement learning)

Basic assumption

We will assume that the plant to be controlled has ***stability properties***, such as ***Input-to-State Stability*** (***ISS)***, which will be formally specified later

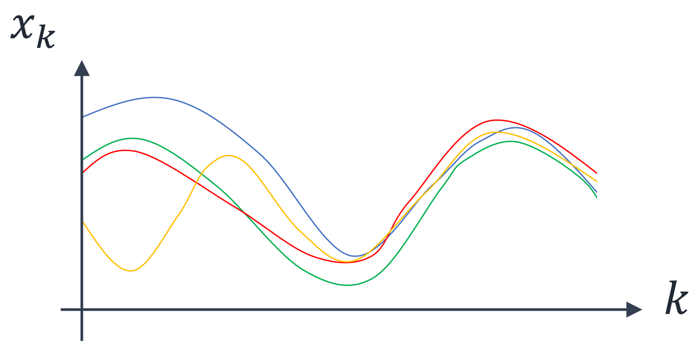
This is a standard assumption, satisfied in many industrial problems

On the other hand, estimation of unstable systems is a really tough problem (pre-stabilization is required)

Outline of the talk

- Motivations and goals for NN-based control
- Recurrent Neural Networks
- **Stability, verification, generalization**
- Physics informed RNN
- Lifelong learning and adaptation
- Conclusions

Preliminaries - Input to State Stability ISS



$$x_{k+1} = f(x_k, u_k)$$

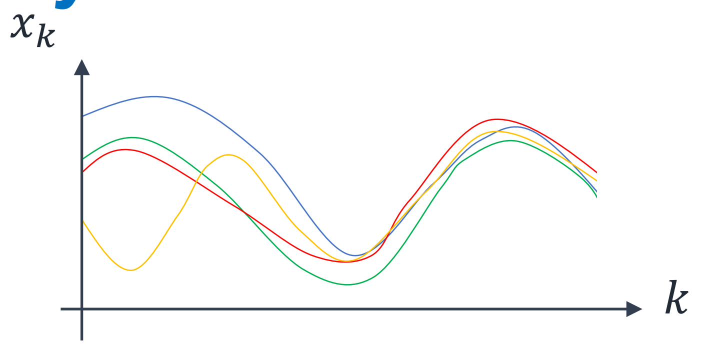
is ISS if $\exists \beta(\|x_0\|, k) \in KL$ and $\gamma_u(\|\mathbf{u}\|_\infty) \in K_\infty$ such that, for any $k > 0$, any initial condition x_0, and any *input sequence* $\mathbf{u}$ it holds that

$$\|x_k(x_0, \mathbf{u})\| \leq \beta(\|x_0\|, k) + \gamma_u(\|\mathbf{u}\|_\infty)$$

The effects of initial conditions asymptotically vanish

The **amplitude of states' trajectories is bounded**

Preliminaries – Incremental Input to State Stability δISS



System

$$x_{k+1} = f(x_k, u_k)$$

is Incrementally ISS if $\exists \beta(\|x_{a0} - x_{b0}\|, k) \in KL$ and $\gamma_u(\|\mathbf{u}_a - \mathbf{u}_b\|_\infty) \in K_\infty$ such that for any $k > 0$, any pair of initial conditions x_{a0}, x_{b0} and any pair of input sequences $\mathbf{u}_a$, $\mathbf{u}_a$ it holds that

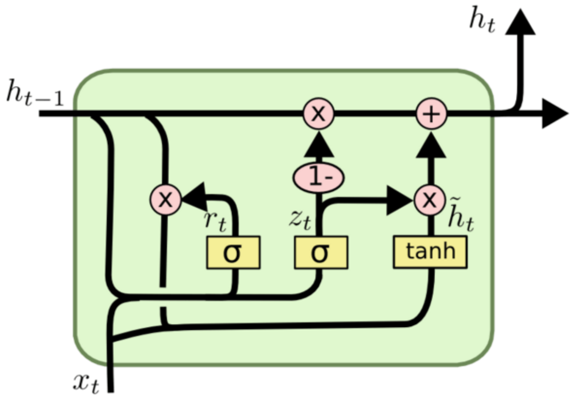
$$\|x_{ak} - x_{bk}\| \leq \beta(\|x_{a0} - x_{b0}\|, k) + \gamma_u(\|\mathbf{u}_a - \mathbf{u}_b\|_\infty)$$

The effects of different initial conditions asymptotically vanish

The **amplitude of the difference of states' trajectories is bounded**

GRU (single layer)

$$
\begin{aligned}
x_{k+1} &= z_k \circ x_k + (1 - z_k) \circ \sigma_c\left(W_r u_k + U_r f_k \circ x_k + b_r\right) \\
z_k &= \sigma_g\left(W_z u_k + U_z x_k + b_z\right) \\
f_k &= \sigma_g\left(W_f u_k + U_f x_k + b_f\right) \\
y_k &= U_o x_k + b_o
\end{aligned}
$$



Infinity norm ISS$_\infty$

$$
\|x(k, x_0, \mathbf{u}, b_r)\|_\infty \leq \beta(\|x_0\|_\infty, k) + \gamma_u(\|\mathbf{u}\|_\infty)
$$

Sufficient condition for ISS$_\infty$

$$
\|U_r\|_\infty \bar{\sigma}_f < 1, \quad \bar{\sigma}_f = \sigma_g\left(\|W_f \quad U_f \quad b_f\|_\infty\right).
$$

δISS of GRU

Infinity norm δISS_∞

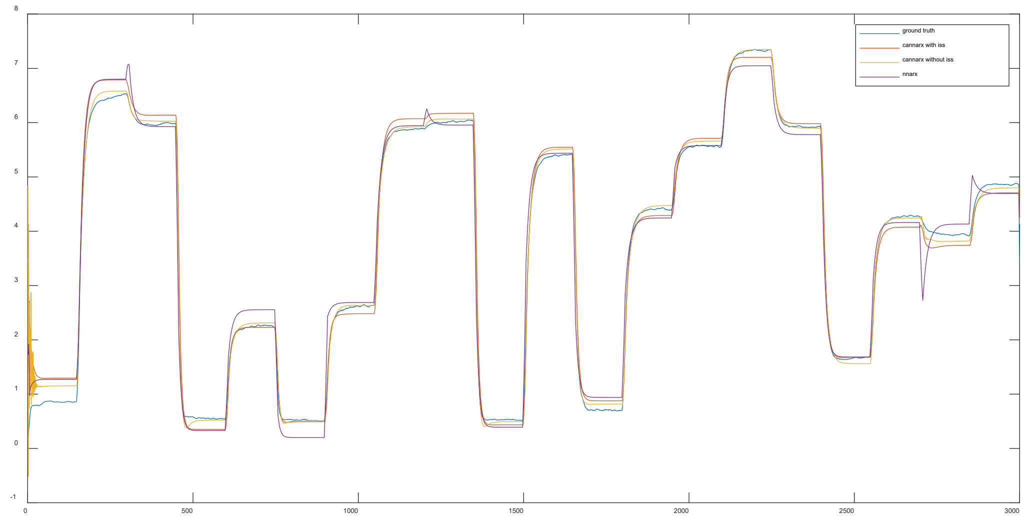
$$\|x(k, x_{a0}, \mathbf{u}_a, b_r) - x(k, x_{b0}, \mathbf{u}_b, b_r)\|_\infty \leq \beta_\Delta(\|x_{a0} - x_{b0}\|_\infty, k) + \gamma_{\Delta u}(\|\mathbf{u}_a - \mathbf{u}_b\|_\infty)$$

Sufficient condition for δISS_∞

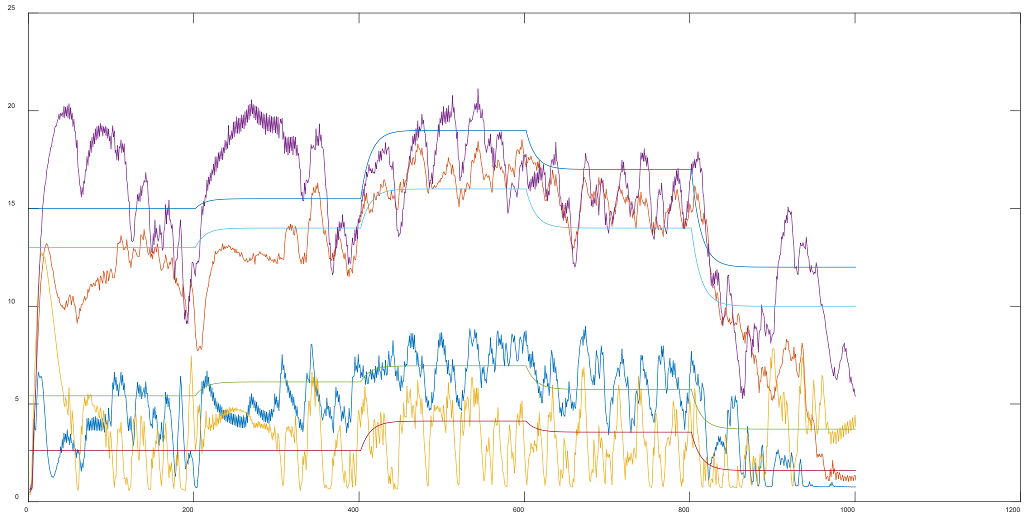
$$\|U_r\|_\infty \left(\frac{1}{4}\|U_f\|_\infty + \bar{\sigma}_f\right) < 1 - \frac{1}{4}\frac{1+\bar{\sigma}_r}{1-\bar{\sigma}_z}\|U_z\|_\infty;$$

$$\begin{aligned} \bar{\sigma}_z &= \sigma_g(\|W_z \quad U_z \quad b_z\|_\infty) < 1, \\ \bar{\sigma}_r &= \sigma_c(\|W_r \quad U_r \quad b_r\|_\infty) < 1 \end{aligned}$$

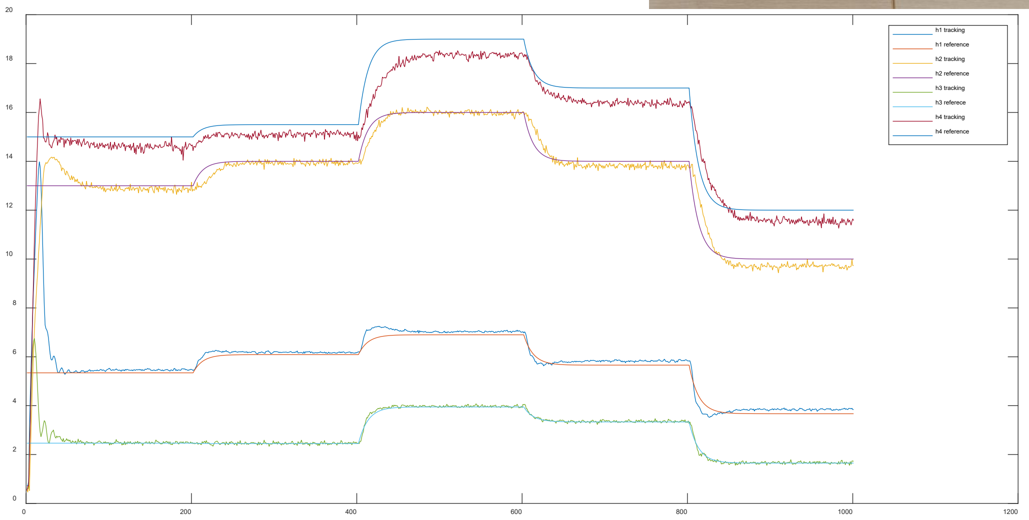
Why ISS and δISS are so important?



Open loop predictions (validation data)



Closed loop (IMC) NO δISS estimated model



Closed loop (IMC) δISS estimated model

Mid-term conclusions…

For all the considered *RNN* structure, strong stability properties can be forced, *ISS* and δ*ISS*

These properties depend upon the model parameters and can be

- Verified a posteriori
- Imposed a priori during the training phase

Warning: only sufficient conditions, could be very conservative (poor approximation performance)

Safety verification

To guarantee that, at least for a given class of inputs, the trained *RNN* does not exhibit meaningless or non-physical outputs

FFNN: reachable sets can be computed unrolling them, *RNN*: open problem

ISS guarantees that, for any initial state and any feasible (bounded) input, the future state trajectories are bounded, but computing the ***analytic bounds*** of the output reachable sets with *ISS* can lead to too tight bounds

A probabilistic solution can be followed based on the ***Scenario Approach***

Analytic bounds for output reachable sets

If a system is ***ISS*** and its output transformation is Lipshitz continuous, it is also ***IOS***, i.e. for any

$x_0 \in X_0$ and any $\mathbf{u} \in U$ it holds that

$$\|y_k(x_0, \mathbf{u})\| \leq \beta_y(\|x_0\|, k) + \gamma_y(\|\mathbf{u}\|_\infty)$$

Now consider a set of candidate inputs $\bar{U} \subseteq U$ and a set of candidate initial states $X_0 \subseteq X$. Then we can analytically compute a set of possible output trajectories

$$\bar{Y}(X_0, \bar{U}) = \{y : \|y\| \leq \beta_y(\sup_{x_0 \in X_0} \|x_0\|, 0) + \gamma_y(\sup_{u \in \bar{U}} \|u\|)$$

This can lead to **very tight explicit bounds** of the Output Reachable Set

Safety verification with the *Scenario approach*

We want to compute the smallest ball Y with radius ρ_y^*, containing the output reachable set.
We reformulate the problem as a chance constrained one: given N sequences (scenarios) of inputs $u^{(i)} \in U$ and initial conditions $x_0^{(i)} \in X_0$, it holds that

$$P_{x,u}\{\|\mathbf{y}(x_0, \mathbf{u})\|_\infty > \rho_y^*(\varepsilon, \beta)\} \leq \varepsilon$$

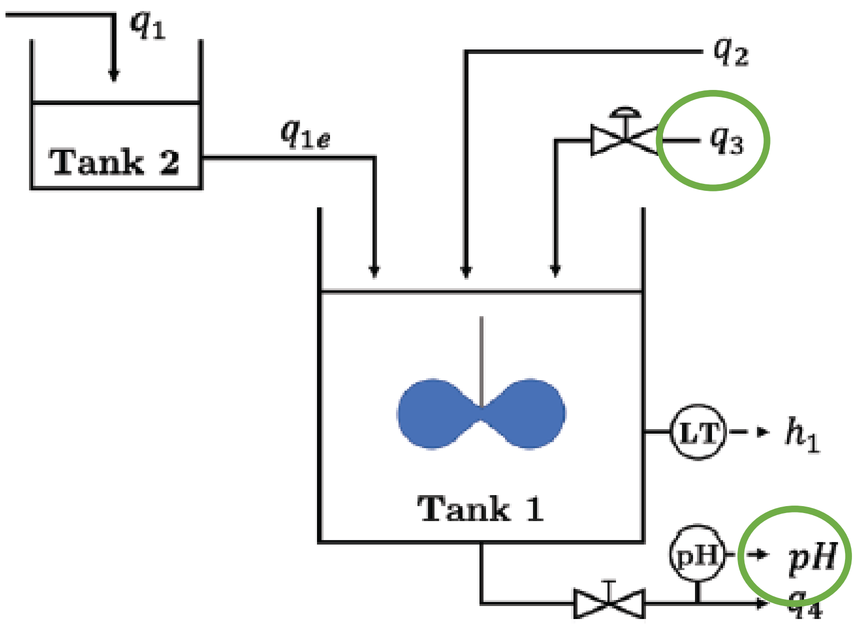
with confidence $1 - \beta$

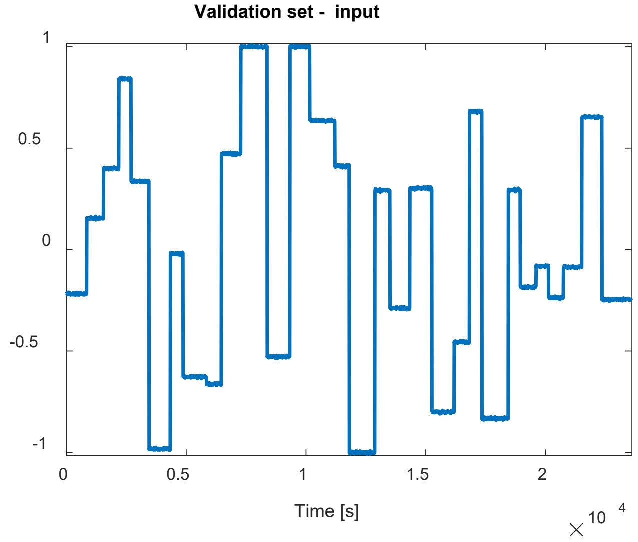
$$\begin{aligned} \rho_y^*(\varepsilon, \beta) &= \min_{\rho_y} \rho_y, \\ \|\mathbf{y}(x_0^{(i)}, \mathbf{u}^{(i)})\|_\infty &\leq \rho_y \qquad \forall i = 1, ..., N \end{aligned}$$

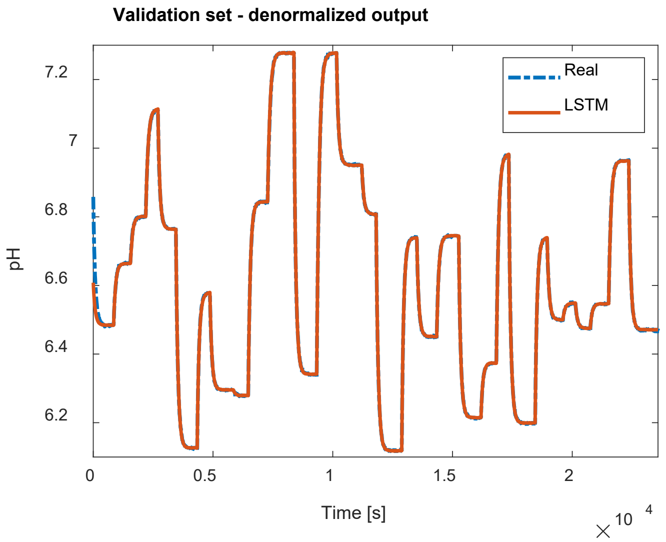
with

$$N \geq \frac{2}{\varepsilon}\left(\ln \frac{1}{\beta} + d\right) \quad , \quad d = 1 \quad n.opt.var.$$

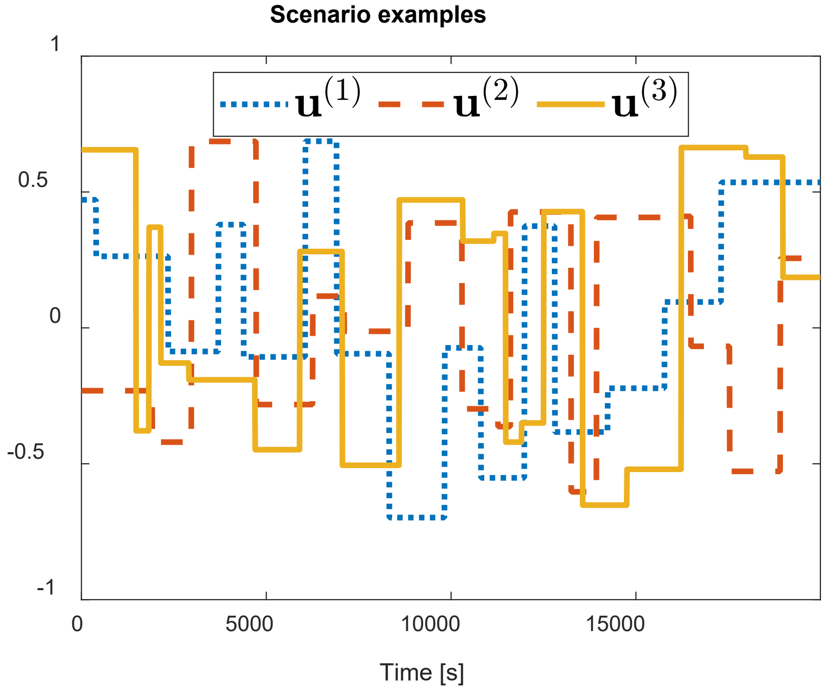
Safety verification – an example

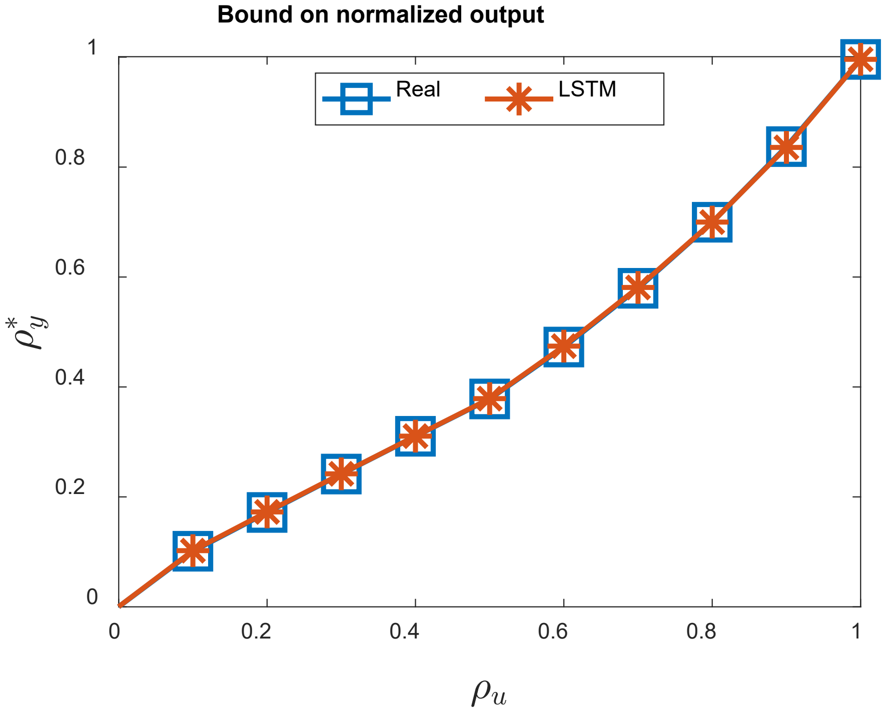






Safety verification – an example





The LSTM is guaranteed to operate almost in the same region of the real system, i.e. a ball with radius $\rho_y^* \simeq 1$

Outline of the talk

- Motivations and goals for NN-based control
- Recurrent Neural Networks
- Stability, verification, generalization
- **Physics informed RNN**
- Lifelong learning and adaptation
- Conclusions

Interpretability - *physics informed NN*

To improve interpretability, and a wide use of RNN, many approaches have been suggested

- *NN* structures enforcing known relations among variables (*T1>T2* at different depths of a lake). Often physical constraints are transformed into soft constraints
- Control affine systems → the NN has a control affine structure
- NN structure with the same topology of (distributed) systems ←

Interpretability – *NN with the same structure of the plant*

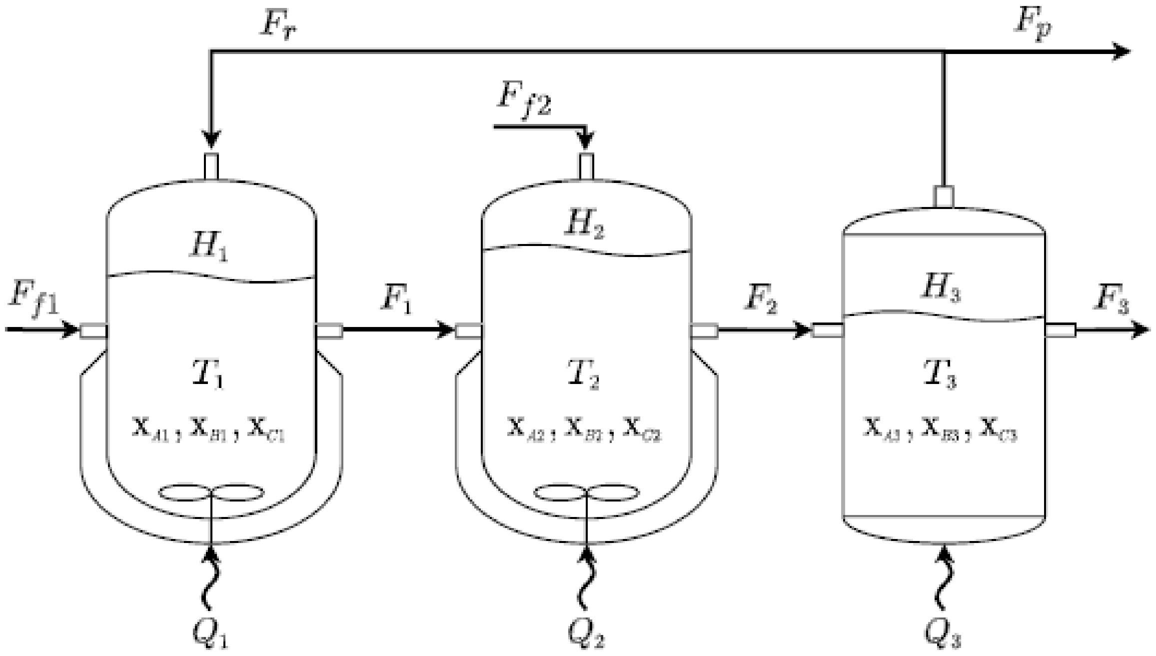


Fig. 3. Two reactors in series with separator and recycle.

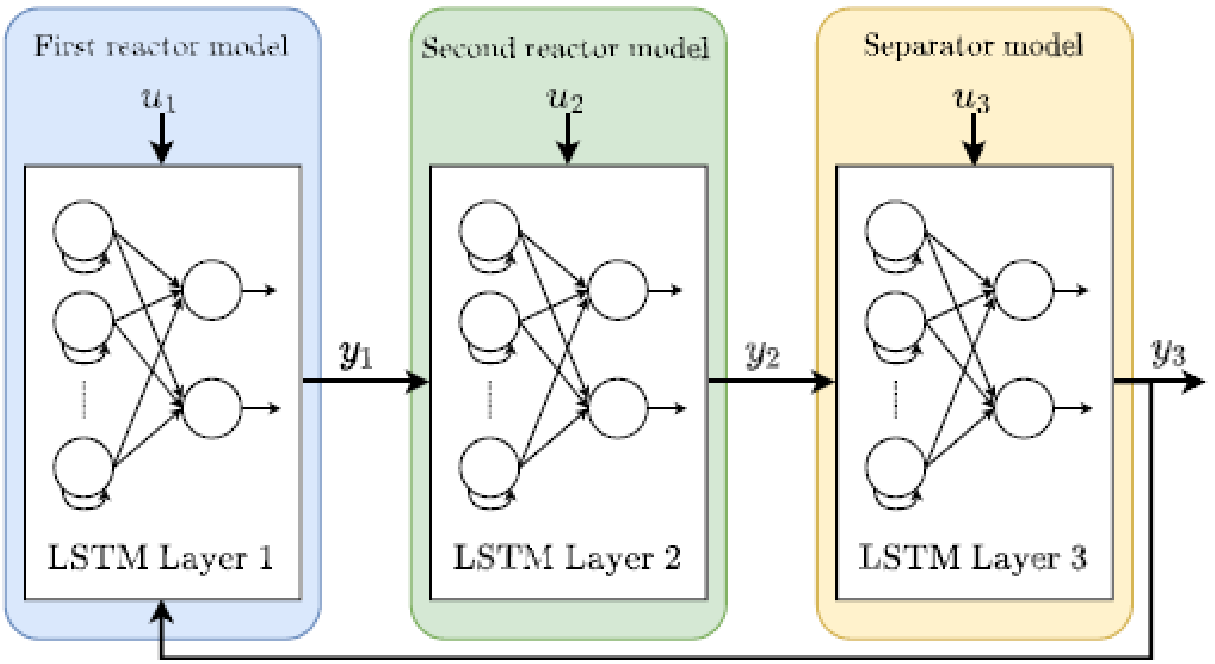
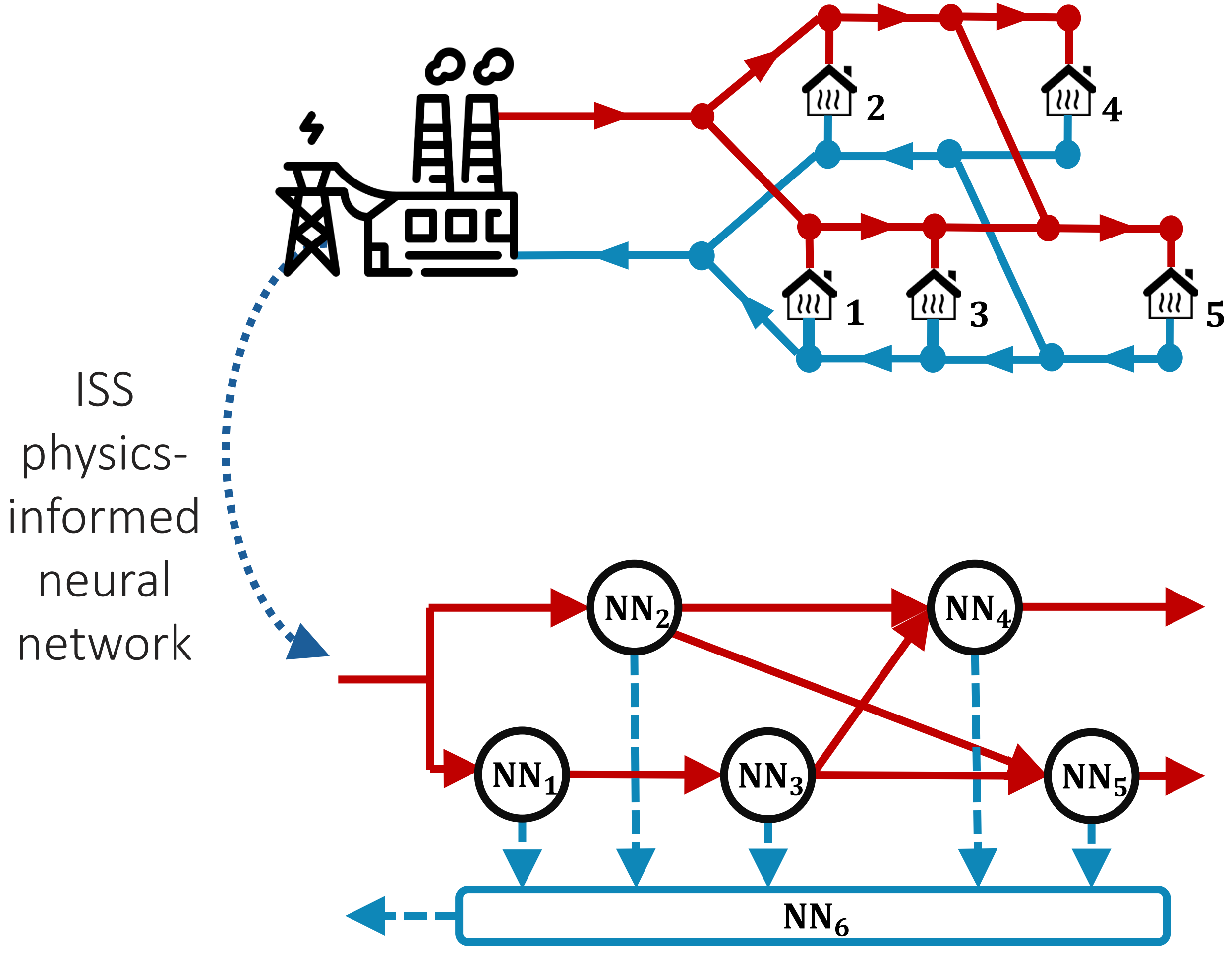


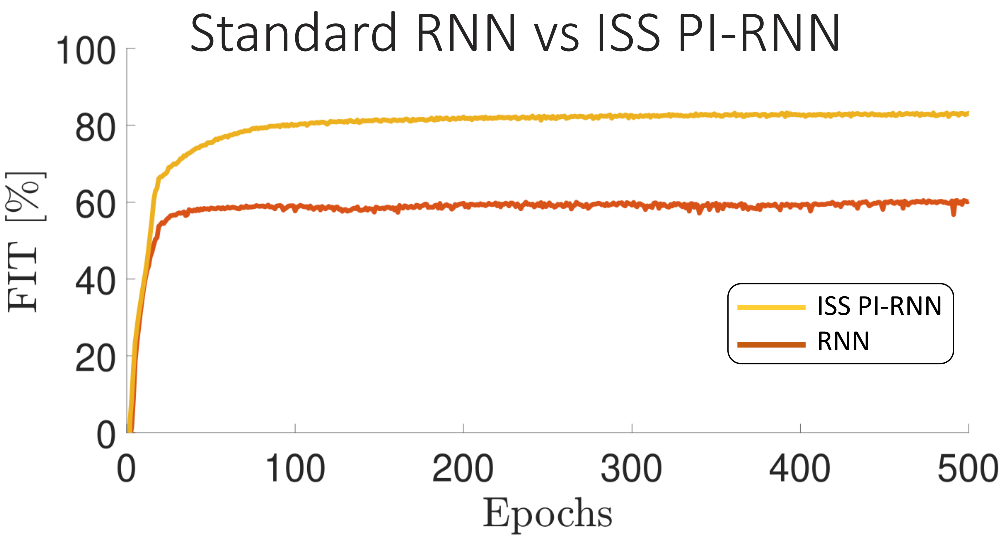
Fig. 4. Structure of the physics-based LSTM used to identify the plant.

Faster and accurate training

ISS of any subsystem implies ISS of the overall one

District heating system



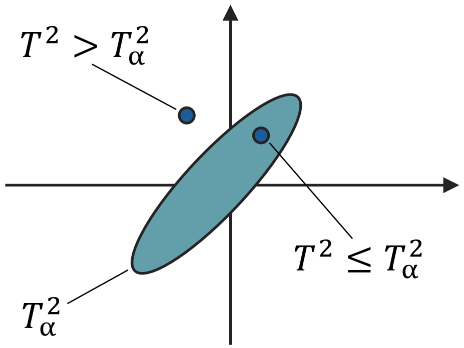


- ✓ Higher identification accuracy
- ✓ Faster training procedure
- ✓ Stability guarantees

Outline of the talk

- Motivations and goals for NN-based control
- Recurrent Neural Networks
- Stability, verification, generalization
- Physics informed RNN
- **Lifelong learning and adaptation**
- Conclusions

Mahalanobis distance



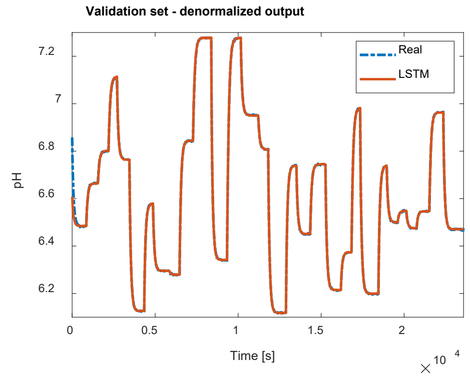
Model monitoring through **multivariate statistical process control**

Anomaly characterization

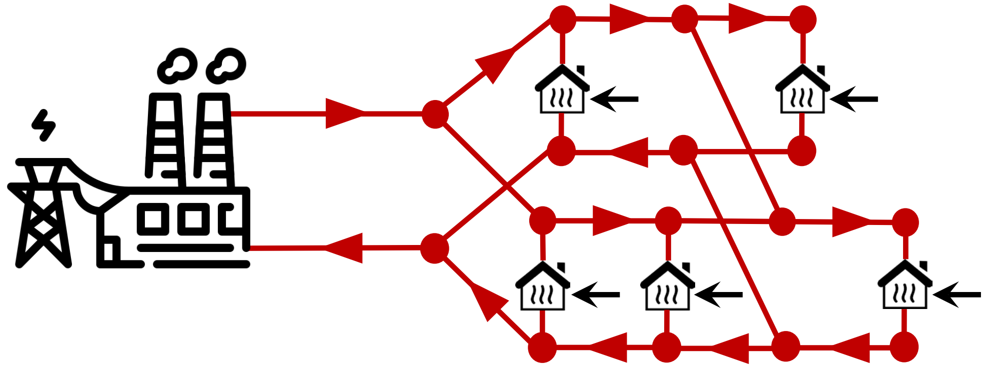
Endogenous change → Model partial update

Exogenous change → Model incremental update

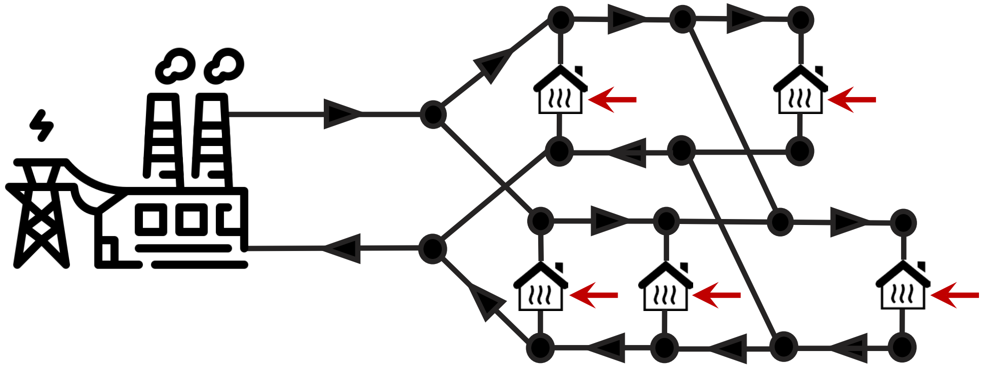
Mean Square Error



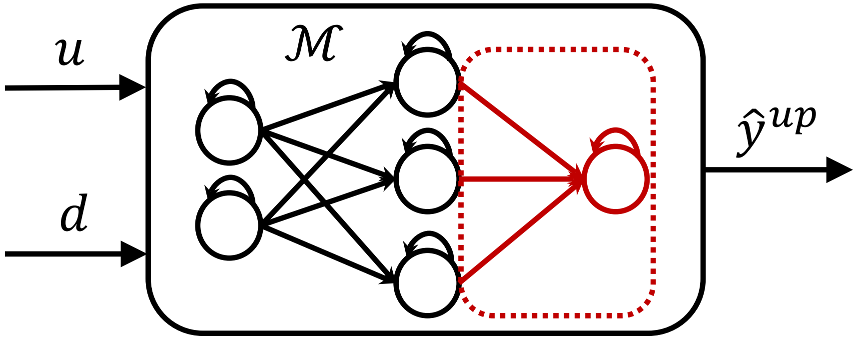
Endogenous system change



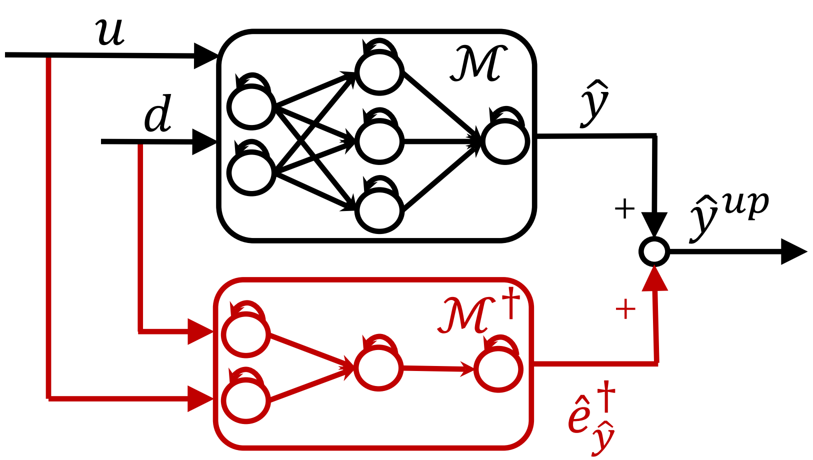
Exogenous system change



Model partial update (MHE)



Model incremental update



Outline of the talk

- Motivations and goals for NN-based control
- Recurrent Neural Networks
- Stability, verification, generalization
- Physics informed RNN
- Lifelong learning and adaptation
- **Conclusions**

Future work

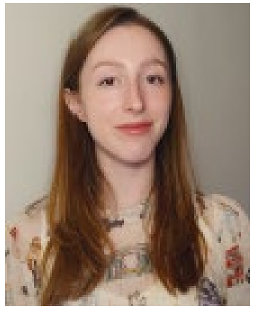
- RNN for unstable systems
- Improving tuning and performance (bootstrap, resampling, …)
- Physics informed: networks of local RNN (local and global ISS, δISS,…)
- Real applications

Thanks to

Fabio Bonassi

Marcello Farina

Luca Bugliari Armenio



Laura Boca de Giuli

Eva Masero Rubio

Alessio La Bella

Thank you