A data-driven surrogate model for uncertainty quantification of dynamical systems

Other Conference Item

Author(s):
Schär, Styfen; Marelli, Stefano; Sudret, Bruno

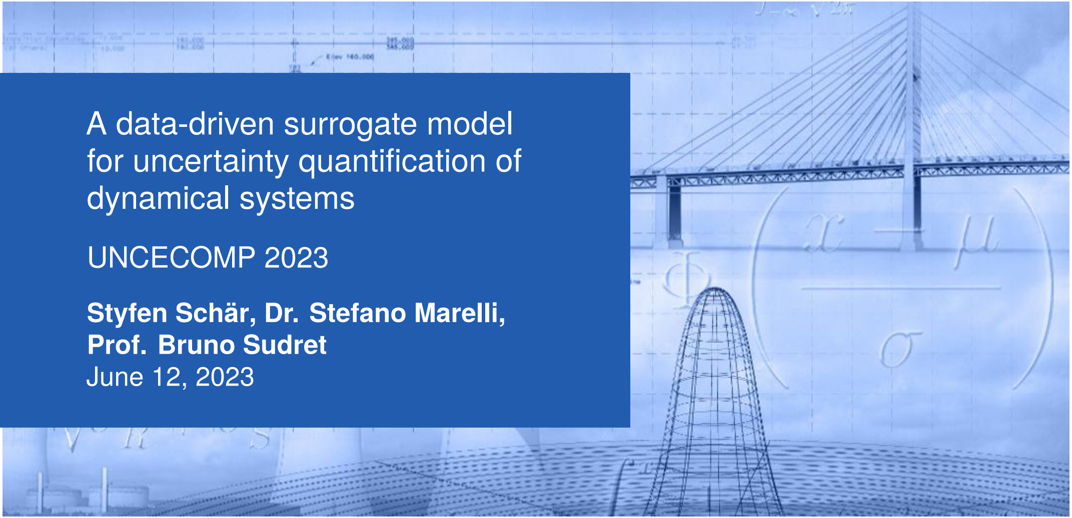
Publication date:
2023-06-12

Permanent link:
https://doi.org/10.3929/ethz-b-000622081

Funding acknowledgement:
101006689 - HIghly advanced Probabilistic design and Enhanced Reliability methods for high-value, cost-efficient offshore WIND (EC)

ETH zürich

Risk, Safety & Uncertainty Quantification



A data-driven surrogate model for uncertainty quantification of dynamical systems

UNCECOMP 2023

Styfen Schär, Dr. Stefano Marelli, Prof. Bruno Sudret

June 12, 2023

Outline

1. Problem statement

2. Surrogate models for dynamical systems

3. Case study

Outline

1. Problem statement
2. Surrogate models for dynamical systems
3. Case study

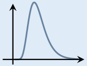
Global framework for uncertainty quantification

Step A
Model(s) of the system
Assessment criteria

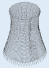
Step B
Quantification of
sources of uncertainty

Step C
Uncertainty propagation

Random variables



Computational model



Distribution
Mean, std. deviation
Probability of failure

Sudret, B. Uncertainty propagation and sensitivity analysis in mechanical models. Habilitation à diriger des recherches, 2007.

- Requires many evaluations of the costly computational model
- Solution: replace the computational model with a cheap-to-evaluate surrogate

Problem setup

Static system

- The surrogate is build from experimental design inputs

$$\mathcal{X} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N_{ED})}\}$$

  and corresponding model outputs

$$\mathcal{Y} = \{\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(N_{ED})}\}$$

- Typically, the surrogate maps a set of scalar inputs to one or more scalar quantities of interest (outputs)

$$\tilde{\mathcal{M}} : \mathcal{D}_{\boldsymbol{x}} \subset \mathbb{R}^M \rightarrow \mathbb{R}$$

Dynamical system

- System has time-dependent exogenous inputs and output

$$\boldsymbol{x} : \mathcal{T} \rightarrow \mathbb{R}^M \text{ and } \boldsymbol{y} : \mathcal{T} \rightarrow \mathbb{R}$$

  that evolve along a time axis

$$\mathcal{T} = \{0, \delta t, 2\delta t, \ldots, N\delta t\}$$

- The surrogate predicts the output at time t based on the input up to and including time t

$$y(t) = \mathcal{M}(\boldsymbol{x}(\mathcal{T} \leq t)) \approx \tilde{\mathcal{M}}(\boldsymbol{x}(\mathcal{T} \leq t))$$

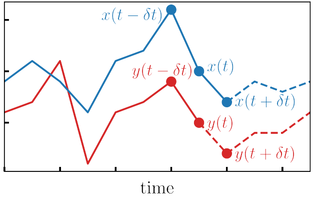
Outline

1. Problem statement
2. Surrogate models for dynamical systems
3. Case study

Surrogate models for dynamical systems

Dynamical systems can often be modelled using AutoRegressive with eXogenous inputs (ARX) models

- Autoregressive: model uses its own past predictions
- Exogenous input: excitation that governs the system response



We denote such models as:

$$y(t) = \mathcal{M}(\boldsymbol{\varphi}(t), \mathbf{c})$$

- $\boldsymbol{\varphi}(t) \in \mathbb{R}^{M_\varphi}$: gathers the exogenous inputs and system output at different time steps
- $\boldsymbol{c}$: finite set of model parameters, e.g. regression coefficients or weights

Calibrating an ARX model

Building an ARX model can be cast into ordinary regression problem $\boldsymbol{y} = \tilde{\mathcal{M}}(\boldsymbol{\Phi}, \boldsymbol{c})$ with regression matrix $\boldsymbol{\Phi}$ and output vector $\boldsymbol{y}$:

$$\boldsymbol{\Phi} = \begin{pmatrix} \boldsymbol{\varphi}(t_0) \\ \boldsymbol{\varphi}(t_0 + \delta t) \\ \vdots \\ \boldsymbol{\varphi}((N-1)\delta t) \end{pmatrix}, \quad \boldsymbol{y} = \begin{pmatrix} \boldsymbol{y}(t_0) \\ \boldsymbol{y}(t_0 + \delta t) \\ \vdots \\ \boldsymbol{y}((N-1)\delta t) \end{pmatrix},$$

where each row $\varphi(\bullet)$ reads

$$\begin{aligned} \boldsymbol{\varphi}(t) = \{ & y(t - \ell_1^y), y(t - \ell_2^y), \dots, y(t - \ell_{n_y}^y), \\ & x_1(t - \ell_1^{x_1}), x_1(t - \ell_2^{x_1}), \dots, x_1(t - \ell_{n_{x_1}}^{x_1}), \\ & x_2(t - \ell_1^{x_2}), x_2(t - \ell_2^{x_2}), \dots, x_2(t - \ell_{n_{x_2}}^{x_2}), \\ & \dots, \\ & x_M(t - \ell_1^{x_M}), x_M(t - \ell_2^{x_M}), \dots, x_M(t - \ell_{n_{x_M}}^{x_M}) \} \end{aligned}$$

- System response $\boldsymbol{y} \in \mathbb{R}^N$
- Exogenous input $\boldsymbol{x} \in \mathbb{R}^{N \times M}$
- Autoregressive lags

$$\ell_i^y \in \{\boldsymbol{\delta t}, 2\delta t, \dots, (N-1)\delta t\}$$

- Exogenous input lags

$$\ell_i^{x_j} \in \{\boldsymbol{0}, \delta t, \dots, (N-1)\delta t\}$$

Challenges with ARX models

Only one timestep at a time:

$$\hat{y}(t) = \hat{\mathcal{M}}(\boldsymbol{x}(\mathcal{T} \leq t), \hat{y}(\mathcal{T} < t))$$

- past predictions $\hat{\boldsymbol{y}}(\mathcal{T} < t)$
- exogenous input $\boldsymbol{x}(\mathcal{T} \leq t)$

Main difficulties

1. Curse of dimensionality
2. Control systems making the response non-smooth
3. Dampers, nonlinear springs and coupling introducing nonlinearities

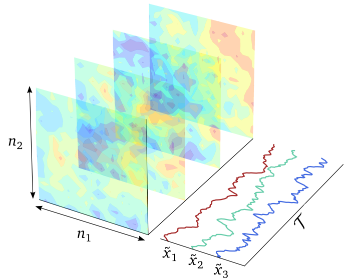
Dealing with high-dimensional exogenous inputs

Reduce dimensionality of the system excitation $\boldsymbol{x}$ along non-temporal coordinates:

$$\tilde{\boldsymbol{x}} = \mathcal{G}(\boldsymbol{x})$$

where $\boldsymbol{x} \in \mathbb{R}^{N \times M}$ and $\tilde{\boldsymbol{x}} \in \mathbb{R}^{N \times m}$ such that $m \ll M$

- Original time scale $\mathcal{T}$ is preserved
- Wide array of methods available, e.g.:
  - n-dimensional discrete cosine transform
  - principal component analysis
  - (deep) autoencoders



Dealing with complex dynamical systems

- Building a surrogate solely on the original exogenous input $\boldsymbol{x} \in \mathbb{R}^{N \times M}$ input can be suboptimal:

$$\tilde{\mathcal{M}} : \boldsymbol{x}(\mathcal{T} \leq t) \rightarrow \boldsymbol{y}(t)$$

- A more informative set of features $\boldsymbol{\zeta} \in \mathbb{R}^{N \times M_\zeta}$ can simplify the mapping to $\boldsymbol{y}$:

$$\tilde{\mathcal{M}} : \boldsymbol{\zeta}(\mathcal{T} \leq t) \rightarrow y(t) \text{ where } \boldsymbol{\zeta} = \mathcal{F}(\boldsymbol{x})$$

- The feature set $\boldsymbol{\zeta}$:
  - does not necessarily have a reduced dimensionality (i.e. $M_\zeta > M$)
  - can and should incorporate prior knowledge about the system

How can we construct and select these features?

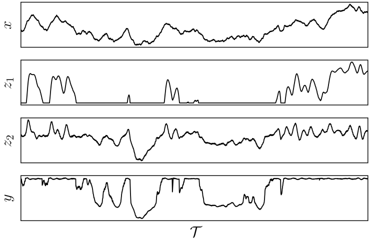
Construction of auxiliary quantities

Auxiliary quantities $\boldsymbol{z}_i$ are constructed incrementally by applying any transform $\mathcal{F}_i$:

$$
\begin{aligned}
\boldsymbol{z}_1(t) &= \mathcal{F}_1(\boldsymbol{x}(\mathcal{T} \leq t), \boldsymbol{z}_1(\mathcal{T} < t)) \\
\boldsymbol{z}_2(t) &= \mathcal{F}_2(\boldsymbol{z}_1(\mathcal{T} \leq t), \boldsymbol{x}(\mathcal{T} \leq t), \boldsymbol{z}_2(\mathcal{T} < t)) \\
&\vdots \\
\boldsymbol{z}_i(t) &= \mathcal{F}_i(\boldsymbol{z}_1(\mathcal{T} \leq t), \ldots, \boldsymbol{z}_{i-1}(\mathcal{T} \leq t), \boldsymbol{x}(\mathcal{T} \leq t), \boldsymbol{z}_i(\mathcal{T} < t))
\end{aligned}
$$

$$\tilde{\mathcal{M}} : \zeta(\mathcal{T} \leq t) \to y(t) \text{ where } \zeta = \{\boldsymbol{x}, \boldsymbol{z}_1, \ldots \boldsymbol{z}_i\}$$

- Transform $\mathcal{F}_i$ can be an ARX model
- An auxiliary quantity can depend on other auxiliary quantities
- Possible auxiliary quantities are e.g.:
  - control system outputs
  - moving averages, integrals/derivates



Automatic selection of auxiliary quantities

```
Function SelectFeatures(x̃, z, y)
    ζ ← {}, ỹ ← y
    while x̃ ≠ {} or z ≠ {} do
        ρ ← Correlate ({x̃, z}, ỹ)
        if max (|ρ|) < θ then
            break
        ζ_i ← arg max(|ρ|)
        if ζ_i ∈ x̃ then
            ζ ← {ζ, ζ_i}
            x̃ ← x̃\ζ_i
        else if ζ_i ∈ z then
            ζ' ← SelectFeatures ({x̃, ζ}, z\ζ_i, ζ_i)
            x̃ ← {x̃, z ∩ ζ'}
            z ← z\z ∩ ζ'
            ζ ← {ζ, ζ_i}
            z ← z\ζ_i
        ỹ ← y − M̃(ζ)
    return ζ

Function Correlate(x, y)
    ρ ← {}
    for x_i ← x_1, ..., x_M do
        ρ ← {ρ, KendallTau(x_i, y)}
    return ρ
```

- $\tilde{\boldsymbol{x}}$: features only depending on exogenous input
- $\boldsymbol{z}$: auxiliary quantities
- $\boldsymbol{y}$: system response
- ζ: selected features
- ρ: measure of association, e.g Kendall's tau

$$\rho(\boldsymbol{x}_i, \boldsymbol{y}) = \frac{2}{n(n-1)} \sum_{k<\ell} \mathrm{sgn}(x_{i,k} - x_{i,\ell})\mathrm{sgn}(y_k - y_\ell)$$

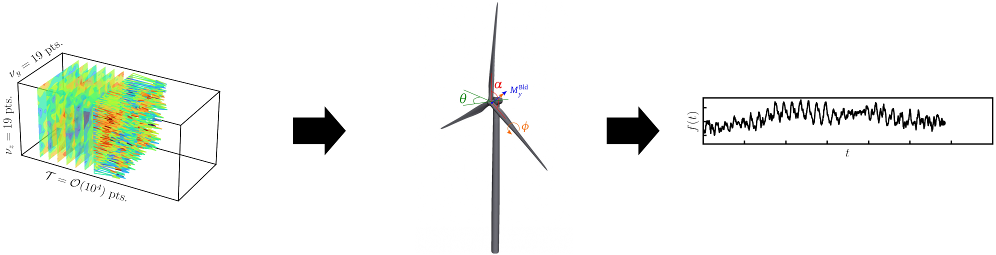
- θ: minimum correlation to stop algorithm
- $\tilde{\mathcal{M}}$: ARX model

Outline

Case study - aero-servo-elastic simulation of a wind turbine

- Input turbulence box $\boldsymbol{v} : \mathcal{T} \rightarrow \mathbb{R}^{\nu_w \times \nu_y \times \nu_z}$
  - Discrete time axis $\mathcal{T} = \{0, \delta t, 2\delta t, \ldots, N\delta t\}$
  - Wind speed components ν_w
  - Discrete spatial grid $\nu_y \times \nu_z$
- Response time series $f : \mathcal{T} \rightarrow \mathbb{R}$

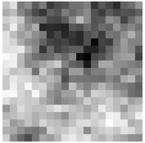
Find a surrogate $\tilde{\mathcal{M}}$ for the aero-servo-elastic wind turbine simulator $\mathcal{M}$:
$f(t) = \mathcal{M}(\boldsymbol{v}(\mathcal{T} \leq t)) \approx \tilde{\mathcal{M}}(\boldsymbol{v}(\mathcal{T} \leq t))$



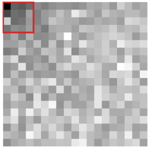
Dimensionality reduction

- Only dominant longitudinal wind speed component $\boldsymbol{v}_x$ is kept
- Decompose each slice $v_x(t)$ into its spectral coefficients $\xi_{i,j}(t)$ using the 2-dimensional discrete cosine transform
- Dimensionality is reduced by choosing $n_i < \nu_y$ and $n_j < \nu_z$

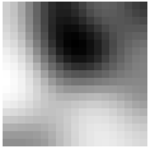
$$v_x^{\kappa,\ell}(t) = \sum_{i=0}^{n_i-1} \sum_{j=0}^{n_j-1} \xi_{i,j}(t) \cos\left[\frac{\pi}{n_i}\left(\kappa+\frac{1}{2}\right) i\right] \cos\left[\frac{\pi}{n_j}\left(\ell+\frac{1}{2}\right) j\right]$$



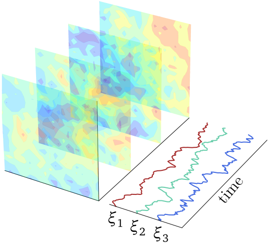
Original frame
$\boldsymbol{v}_x(t=t^*)$



Spectral coefficients
$\log|\boldsymbol{\xi}(t=t^*)|$



Reconstructed frame
$\tilde{\boldsymbol{v}}_x(t=t^*)$



Time-dependent spectral coefficients

Feature selection

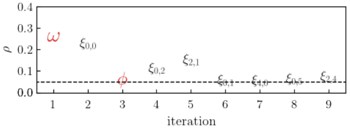
- Quantity of interest: Power output $\boldsymbol{P}$
- Pool of features
  - Spectral coefficients $\boldsymbol{\xi}_{0,0}, \ldots, \boldsymbol{\xi}_{6,6}$
  - Blade pitch $\boldsymbol{\phi}$
  - Rotor speed $\boldsymbol{\omega}$
- Stopping criterion: $\rho < \theta = 0.05$
- Using Polynomial NARX model:

$$y(t) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} c_{\boldsymbol{\alpha}} \mathcal{P}_{\boldsymbol{\alpha}}(\boldsymbol{\varphi}(t))$$

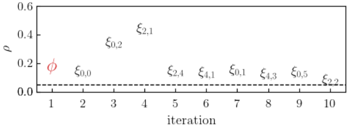
where

$$\mathcal{P}_{\boldsymbol{\alpha}}(\boldsymbol{\varphi}(t)) = \prod_{i=1}^{M_{\varphi}} \boldsymbol{\varphi}_i(t)^{\alpha_i}$$

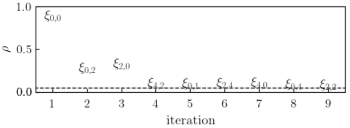
Power output $\boldsymbol{P}$



Rotor speed $\boldsymbol{\omega}$



Blade pitch $\boldsymbol{\phi}$

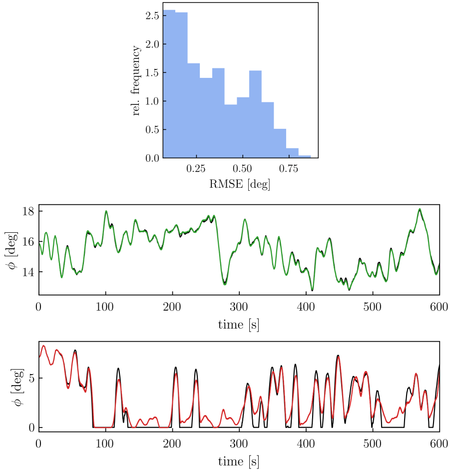


Results - Blade pitch

Auxiliary quantity

$$\hat{\phi}(t) = \tilde{\mathcal{M}}(\boldsymbol{\xi}(\mathcal{T} \leq t), \hat{\phi}(\mathcal{T} < t))$$

- Root-mean-squared (RMSE) error on the out-of-sample validation dataset
- Simulation with lowest and highest RMSE
- Surrogate shows high accuracy
- Surrogate is slightly less responsive than the turbine controller

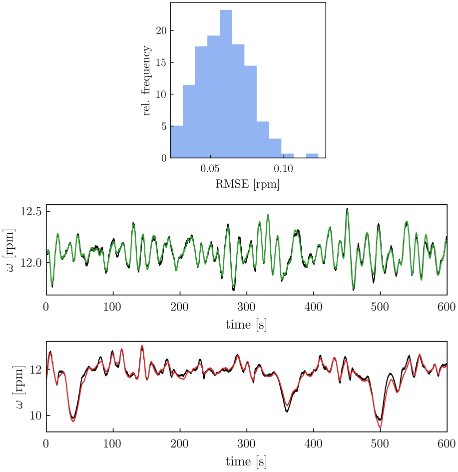


Results - Rotor speed

Auxiliary quantity

$$\hat{\omega}(t) = \tilde{\mathcal{M}}(\boldsymbol{\xi}(\mathcal{T} \leq t), \hat{\phi}(\mathcal{T} \leq t), \hat{\omega}(\mathcal{T} < t))$$

- Surrogate shows high accuracy
- Prediction is missing very high-frequency components
- Prediction remains stable also when relying on the blade pitch prediction

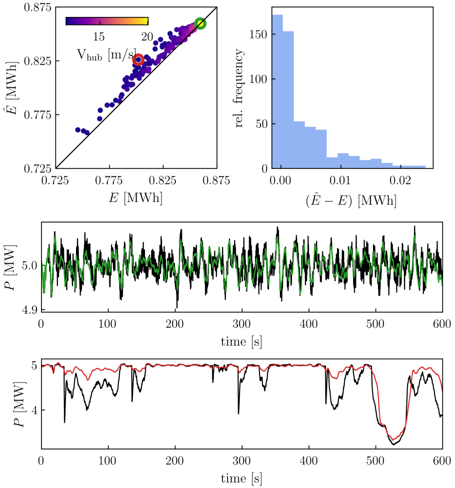


Results - Generator power

Quantity of interest

$$\hat{P}(t) = \tilde{\mathcal{M}}(\boldsymbol{\xi}(\mathcal{T} \le t), \hat{\phi}(\mathcal{T} \le t), \hat{\omega}(\mathcal{T} \le t), \hat{P}(\mathcal{T} < t))$$

- Surrogate tends to slightly overestimate power output (< 2 % in average)
- Sharp dips are not captured well by the surrogate
- Accurate surrogate over the full 10 minutes, even when relying on blade pitch and rotor speed prediction



Conclusion and outlook

Conclusion

- Multistep surrogate modelling approach can be favourable as it simplifies the input-output mapping at each step
- The final surrogate prediction is stable over extended time periods, even when relying on auxiliary quantities
- Construction order of auxiliary quantities can be determined in a data-driven recursive fashion

Outlook

- Investigate the effect of different stopping criteria and measures of association
- Application to other problems, such as buildings under wind loads and ground motion

Chair of Risk, Safety & Uncertainty Quantification

www.rsuq.ethz.ch

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 101006689

The Uncertainty Quantification Software

www.uqlab.com

UQLab

www.uqpylab.uq-cloud.io

UQCloud

The Uncertainty Quantification Community

www.uqworld.org

UQWorld