Echo-state-network-based iterative learning control of distributed systems

The 22nd World Congress of the International Federation of Automatic Control
Yokohama, Japan, July 9-14

Krzysztof Patan, Maciej Patan

Inst. of Control and Computation Engn.
University of Zielona Góra, Poland

Introduction

- Distributed-Parameter System (DPS) control – a challenging task, especially for non-linear systems
  - linearization – problems in the case of non-homogeneous systems
  - lumping – information loss
- Finite Element Method (FEM) – an established approach to deal with DPS
  - dense mesh required
    - large computational burden
    - possible problems with numerical stability and convergence
  - off-line procedure
- Alternative solution – to employ neural network models to represent DPS
- **Objective** – to apply echo-state network to design iterative learning control for a class of nonlinear distributed-parameter systems

Distributed-parameter system representation

Let us consider the system

$$\frac{\partial y}{\partial t} = \mathcal{F}\big(x, t, y, \nabla y, \nabla^2 y; u\big), \quad (x,t) \in \Omega \times T, \tag{1}$$

subject to the boundary and initial conditions

$$\mathcal{B}\big(x, t, y, \nabla y; u\big) = 0, \quad (x,t) \in \partial\Omega \times T, \tag{2}$$

$$y\big(x, 0\big) = y_0(x), \qquad x \in \Omega, \tag{3}$$

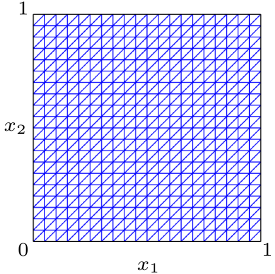
where $y = y(x,t)$ – the system state at the point x of the spatial domain $\Omega \in \mathbb{R}^2$

$\mathcal{B}$, $\mathcal{F}$, y_0 – known non-linear functions

∇ and ∇^2 – the gradient and Hessian, respectively

u – the vector of system actuating inputs

- Common approach – Finite Element Method
- Partition of the clamped plate by FEM



- Dense spatial discretization is required to assure high accuracy of solution
- Many nodes, e.g. 441 nodes $\rightarrow$ 800 triangles
- DPS solving – complex method with high computation time

Echo-state neural network

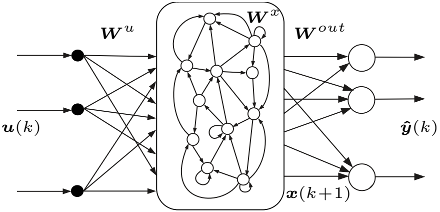
- Machine learning method for mapping inputs into a high dimensional space
- The key idea – using a reservoir of non-linear processing units
- Processing units are connected using recurrent links
- Let us consider the model

$$\boldsymbol{x}(k+1) = \boldsymbol{f}_h\left(\boldsymbol{W}^x \boldsymbol{x}(k) + \boldsymbol{W}^u \boldsymbol{u}(k+1)\right),$$
$$\hat{\boldsymbol{y}}(k) = \boldsymbol{f}_o\left(\boldsymbol{W}^{out} \boldsymbol{x}(k)\right),$$

where $\boldsymbol{x}(k)$, $\boldsymbol{u}(k)$, $\hat{\boldsymbol{y}}(k)$ – the state, input and output vectors
$\boldsymbol{W}^x$, $\boldsymbol{W}^u$, $\boldsymbol{W}^{out}$ – the reservoir, input and the output weight matrices
$\boldsymbol{f}_h$, $\boldsymbol{f}_o$ – activation functions of hidden and output neurons

- Weight matrices $\boldsymbol{W}^x$ and $\boldsymbol{W}^u$ are chosen randomly
- $\boldsymbol{W}^x$ is sparse (a few % of connections)

Network structure



Training process

$$\boldsymbol{W}^{out} = ((\boldsymbol{X} + \mu \mathbb{I})^{-} \boldsymbol{Y})^{\mathsf{T}},$$

where $\boldsymbol{X}$, $\boldsymbol{Y}$ – state and teacher output collection matrices

μ – regularization parameter, $\mathbb{I}$ – the identity matrix

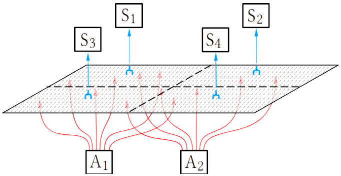
Echo-state property

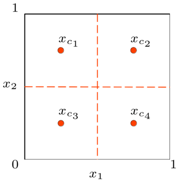
$$\sigma_{\max}(\boldsymbol{W}^x) < 1,$$

where σ_{max} – the maximum singular value

Actuating and sensing

- To reduce data needed for identification the spatial domain is split into smaller regular areas
- To each area a sensor measuring the output is assigned $S_i,\ i = 1, \ldots, n_s$
- The spatial area is actuated using a number of actuators $A_i,\ i = 1, \ldots, n_a$

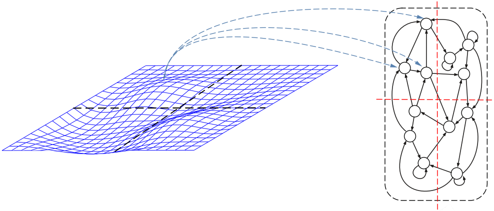




- Point-wise measurements are acquired $z^i(t) = y(x_{c_i}, t)$
- Data recorded by all sensors – **output patterns**
- Excitation of the system – a number of actuators
- Data provided by all actuators – **input data**

Spatial reservoir

- Identification goal – learn both the spatial and time systems characteristics
- Our proposition – to divide the reservoir into smaller sub-regions which are fed with suitable point-wise actuation
- Units are sparsely connected
- Each partition consists of n_p units
- Each spatial variable was divided into R sectors
- R^2 – the number of partitions



- Units in the partition are excited only by a suitable actuations
- Number of units in the reservoir: $R^2 \times n_p$

Control scheme

- Data-driven iterative learning control

$$\boldsymbol{u}_{p+1}(k) = \boldsymbol{u}_p(k) + L\boldsymbol{e}_p(k)$$

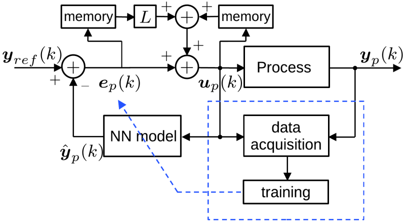
where

p – the trial

k – the time instant

L – a learning gain

$\boldsymbol{e}_p(k) = \boldsymbol{y}_{ref}(k) - \hat{\boldsymbol{y}}_p(k)$ – the tracking error



- Learning gain – scalar value

1. experimentally developed

$$L(p) = -(17000e^{-0.02p} + 3000) \tag{4}$$

2. Xu and Tan (2003)

$$L(p) = \frac{2}{\alpha_1 + \alpha_2}, \quad 0 < \alpha_1 < \frac{\partial \boldsymbol{f}_o}{\partial \boldsymbol{u}_p} \leq \alpha_1 \tag{5}$$

Illustraive example

Clamped elastic membrane

$$\rho\frac{\partial^2 y(x,t)}{\partial t^2} + \kappa\nabla^4 y(x,t) = u(x,t), \quad \kappa = \frac{Ed^3}{12(1-\nu^2)}$$

where $y(x,t)$ – transverse displacement, $u(x,t)$ – pressure field, x – a spatial point, t – time
$\rho = 2700$ – the mass density $E = 7.11 \cdot 10^{10}$ – the elasticity modulus
$\nu = 0.3$ – the Poisson's ratio, $d = 0.003m$ – the plate thickness

the initial conditions at the boundary $\partial\Omega$:

$$y(x,t) = 0 \quad x \in \partial\Omega$$

the initial conditions:

$$y(x,0) = 0, \quad \dot{y}(x,0) = 0, \quad x \in \Omega$$

Reference displacement

- elliptic paraboloid profile

$$y_{ref}(t) = 10^{-3}\left(1 - \frac{|t-100|}{100}\right) e^{-20\left((x_1-0.4)^2+(x_2-0.6)^2\right)}.$$

- the length of each trial: $20s$
- the sampling time: $T_s = 0.1s$
- the length of the reference (number of samples): 201

Model design

- spatial variable range division: $R = 5$
- the number of model inputs and outputs: $R^2 = 25$
- the size of the reservoir partition: $n_p = 10 \rightarrow 250$ units in the reservoir
- the neurons connection sparsity ratio: 20%
- the largest singular value of the state matrix: $\sigma_{max} = 0.95$
- activation functions: $\boldsymbol{f}_o$ – linear, $\boldsymbol{f}_h$ – hyperbolic tangent
- input and output data were scalled to the interval $[-10, 10]$
- training with regularization: $\mu = 0.1$

Data gathering: an excitation was applied at different locations of the plate and the plate displacement was recorded by sensors: Patan and Patan, 2022, Reservoir modeling of distributed-parameter systems, ICARCV 2022

ILC design

- ESN model was used to predict the plate displacement
  - after each trial the model parameters were fine-tuned using the rule

$$\boldsymbol{W}^{out}_{new} = \boldsymbol{W}^{out}_{old}(1-\lambda) + \boldsymbol{W}^{out}\lambda \quad (6)$$

  where λ – a forgetting parameter $\lambda \in (0,1)$

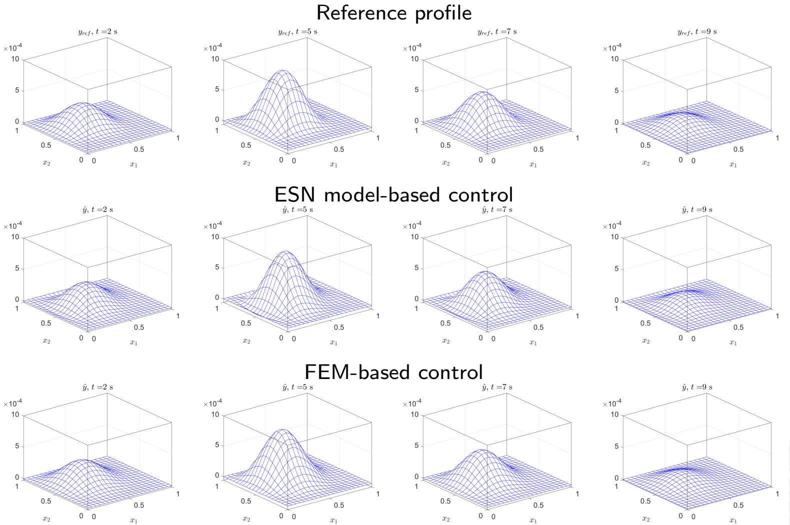
  $\boldsymbol{W}^{out}$ – calculated using the fundamental learnig rule

  - in our study: $\lambda = 0.05$

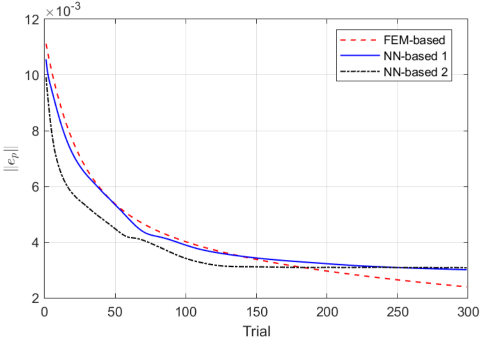
- comparative evaluation: FEM based ILC
  - spatial grid: 21×21, 441 nodes, 800 spatial regions
  - the learning gain: $L(p) = -(5000e^{-0.012p} + 2500)$

- performance index: norm of the tracking error

Control results



ILC Convergence



Gains:
FEM-based: $L(p) = -(5000e^{-0.012p} + 2500)$
NN-based1: $L(p) = -(5000e^{-0.012p} + 2500)$
NN-based2: $L(p) = 0.1\frac{2}{\alpha}, \alpha = \max\{\boldsymbol{W}^{out}\boldsymbol{W}^{u}\}$

Concluding remarks

- Proposed data-driven ILC is less computationally expensive than FEM-based one
  - ESN-ILC was three times faster than FEM-based approach
- Real-time properties of the developed control scheme
- Network parameters can be adapted on demand, not after each operation cycle
- Future research directions:
  - to select more accurately the learning gain
  - to perform convergence analysis of the control scheme
  - optimal selection of sub-regions