St Petersburg University
www.spbu.ru

Neural Networks in Modeling Beam Dynamics using Taylor Mapping

Nataliia Kulabukhova, Anna Golovkina, Vladimir Kozynchenko

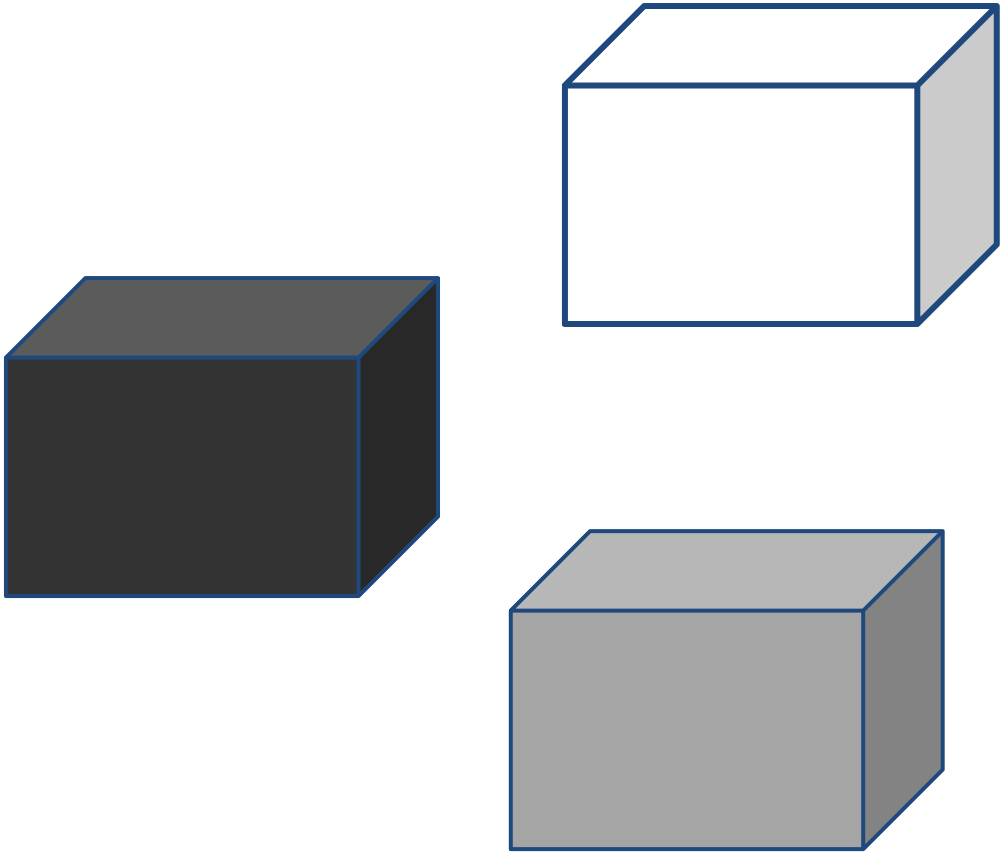
Saint Petersburg State University

GRID 2021 Online, 06.07.2021

Survey

Dynamical system reconstruction and identification:

- white-box;
- grey-box;
- black-box.



Problems

- require large volumes of measured or simulated data;
- building surrogate models that can replace physics-based models;
- use various architectures of NN for model-free systems learning and control;
- do not consider generalisation ability of the constructed models for the inputs not presented in the training data range.

Idea

First. Approach suitable for identification of wide variety of dynamical process with small training data.

Second. The recovered model is requested to predict the dynamics for new inputs beyond the training samples.

Polynomial Neural Network (PNN)

PNN Algorithm

First. Reconstruction of system of ODE with a polynomial right side with only one sample.

Second. Taylor mapping representation of the general solution.

Third. Initialization of PNN weights based on general solution representation.

PNN Architecture

$$\mathcal{M} : \mathrm{X}_0 = \mathrm{X}(t_0) \to \mathrm{X}(t_1)$$

$$\mathrm{X}(t_1) = W_0 + W_1 \mathrm{X}_0 + W_2 \mathrm{X}_0^{[2]} + \ldots + W_k \mathrm{X}_0^{[k]}, \qquad (1)$$

where $\mathrm{X}, \mathrm{X}_0 \in R^n$, matrices W_i are weights, and $\mathrm{X}^{[k]}$ means the kth Kronecker power of vector X with the same terms reduction.

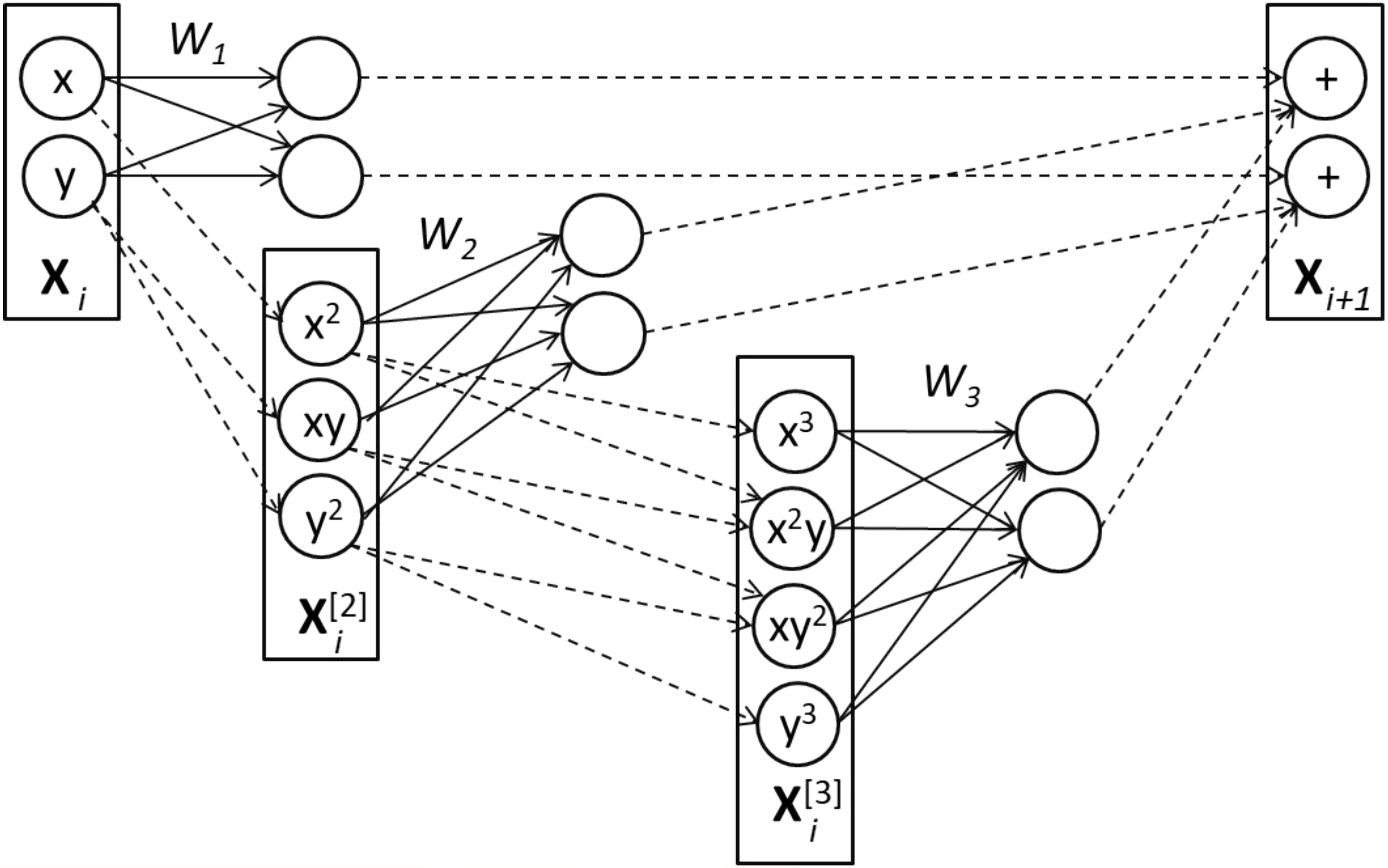
$$\mathrm{X} = (x_1, x_2)$$

Kronecker powers if k = 1, 2, 3

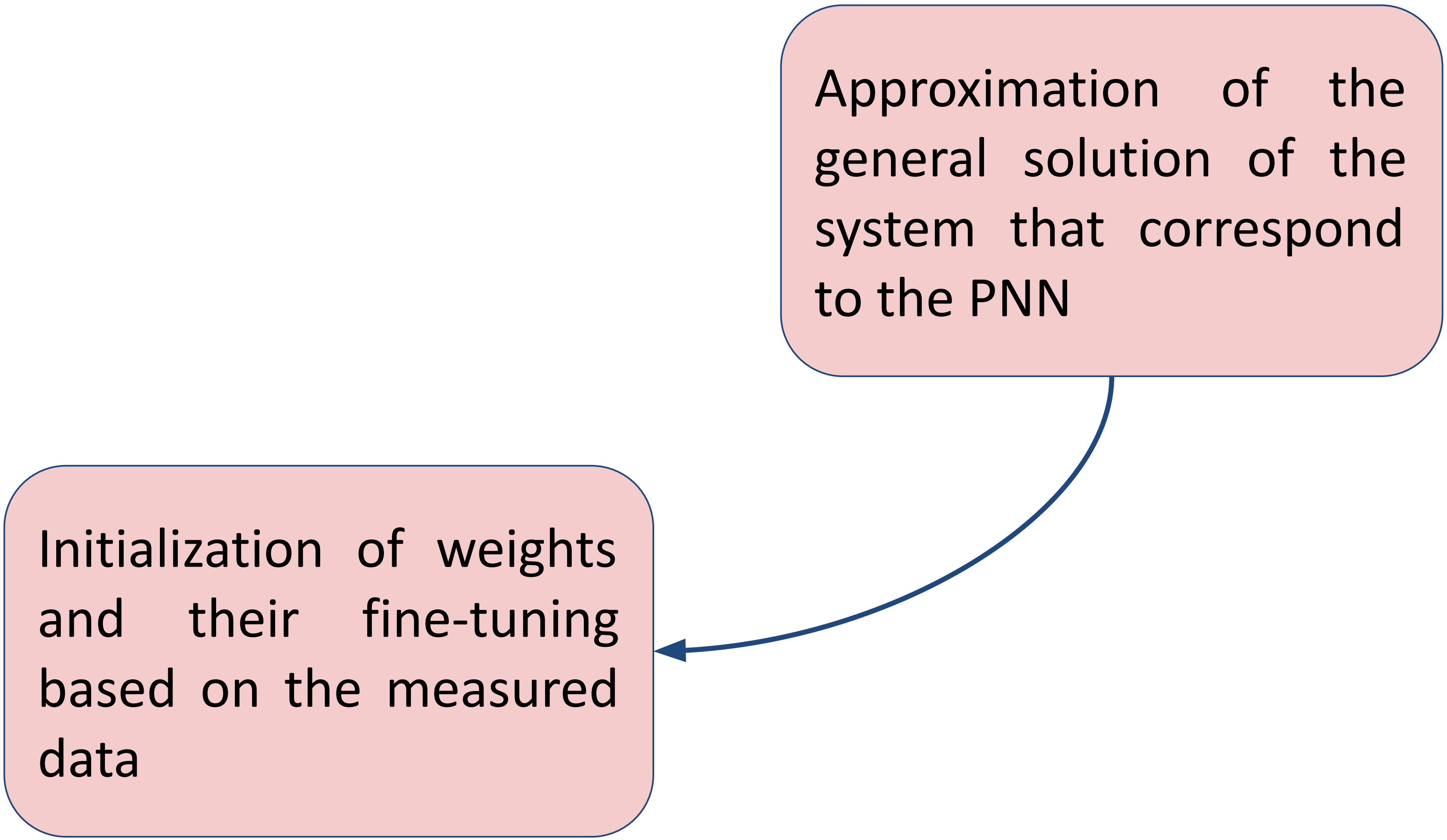
$$\mathrm{X}^{[2]} = (x_1^2, x_1 x_2, x_2^2)$$

$$\mathrm{X}^{[3]} = (x_1^3, x_1^2 x_2, x_1 x_2^2, x_2^3)$$

PNN Architecture



PNN Learning



Reconstruction of the right side

$$X_j(t), j = \overline{1, n}$$

$$t_0, \ldots, t_{M+1} : \mathrm{X}(t_0), \ldots \mathrm{X}(t_{M+1})$$

$$\frac{d\mathrm{X}}{dt} = \sum_{k=0}^{N} \mathrm{P}^{1k} \mathrm{X}^{[k]}$$

$$\frac{\mathrm{X}(t_{i+1}) - \mathrm{X}(t_{i-1})}{t_{i+1} - t_{i-1}} = \sum_{k=0}^{N} \mathrm{P}^{1k} \mathrm{X}^{[k]}(t_i), \quad i = \overline{1, M}.$$

$$AP = B,$$

Reconstruction of the right side

$$AP = B,$$

where:

$$A = \begin{pmatrix} \mathbf{Y}(t_1) & \mathbf{Y}^{[2]}(t_1) & \dots & \mathbf{Y}^{[N]}(t_1) \\ \mathbf{Y}(t_2) & \mathbf{Y}^{[2]}(t_2) & \dots & \mathbf{Y}^{[N]}(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Y}(t_M) & \mathbf{Y}^{[2]}(t_M) & \dots & \mathbf{Y}^{[N]}(t_M) \end{pmatrix}, \quad \mathbf{Y}^{[j]}(t_i) = \left(\mathbf{X}^{[j]}(t_i)\right)^T,$$

$$P = \left(\mathrm{P}^{11}, \dots, \mathrm{P}^{1N}\right)^T,$$

$$B = \left(\left(\mathbf{X}(t_2) - \mathbf{X}(t_0)\right)/(t_2 - t_0), \dots, \left(\mathbf{X}(t_{M+1}) - \mathbf{X}(t_{M-1})\right)/(t_{M+1} - t_{M-1})\right)^T$$

Identification of the general solution

Linear equation:

$$\frac{d\mathbf{X}}{dt} = P^{11}\mathbf{X}$$

The solution to this equation:

$$\mathrm{R}^{11}(t, t_0) = \exp\left((t - t_0)\,\mathrm{P}^{11}\right)$$

$$\mathrm{X}(t) = \mathrm{R}^{11}(t, t_0)\,\mathrm{X}_0, \qquad \mathrm{X}_0 = \mathrm{X}(t_0)$$

Nonlinear equation with the second order polynomials:

$$\frac{d\mathbf{X}}{dt} = \mathrm{P}^{11}\mathbf{X} + \mathrm{P}^{12}\mathbf{X}^{[2]}.$$

$$\mathbf{X}(t) = \mathrm{R}^{11}(t, t_0)\,\mathbf{X_0} + \mathrm{R}^{12}(t, t_0)\,\mathbf{X_0}^{[2]},$$

…

Example. Cylindrical deflector

$$x' = y$$
$$y' = -2x + x^2/R$$

- initial condition - $X_0 = (-2, 4)$
- The training set includes 27 points ($R = 10$)

$$\frac{d\mathbf{X}}{dt} = \begin{pmatrix} 4.06732625 \cdot 10^{-6} & 0.999964923 \\ -1.99993049 & -4.38301027 \cdot 10^{-6} \end{pmatrix} \mathbf{X} +$$

$$\begin{pmatrix} 1.24062940e \cdot 10^{-7} & 3.28166404 \cdot 10^{-6} & 2.00339676 \cdot 10^{-9} \\ 0.0999900634 & -9.71187750 \cdot 10^{-8} & 3.33946651 \cdot 10^{-6} \end{pmatrix} \mathbf{X}^{[\mathbf{2}]} +$$

$$10^{-7} \begin{pmatrix} -3.26578011 & 1.48318573 & -1.79415514 & 0.72283912 \\ 1.07008208 & 3.66009614 & -1.31485102 & 1.79102455 \end{pmatrix} \mathbf{X}^{[\mathbf{3}]} \quad (22)$$

Example. Cylindrical deflector

$$\mathbf{X}\left(t=1.1\right)=\begin{pmatrix} 0.0152181223886654 & 0.707024692103253 \\ -1.41404983954638 & 0.0152121475825109 \end{pmatrix}\mathbf{X_0}+$$

$$\begin{pmatrix} 0.0330738670919230 & 0.0232052217767992 & 0.00808282751349695 \\ 0.0242876681630190 & 0.0338195709561698 & 0.0232073819255883 \end{pmatrix}\mathbf{X_0^{[2]}}$$

Example. Cylindrical deflector

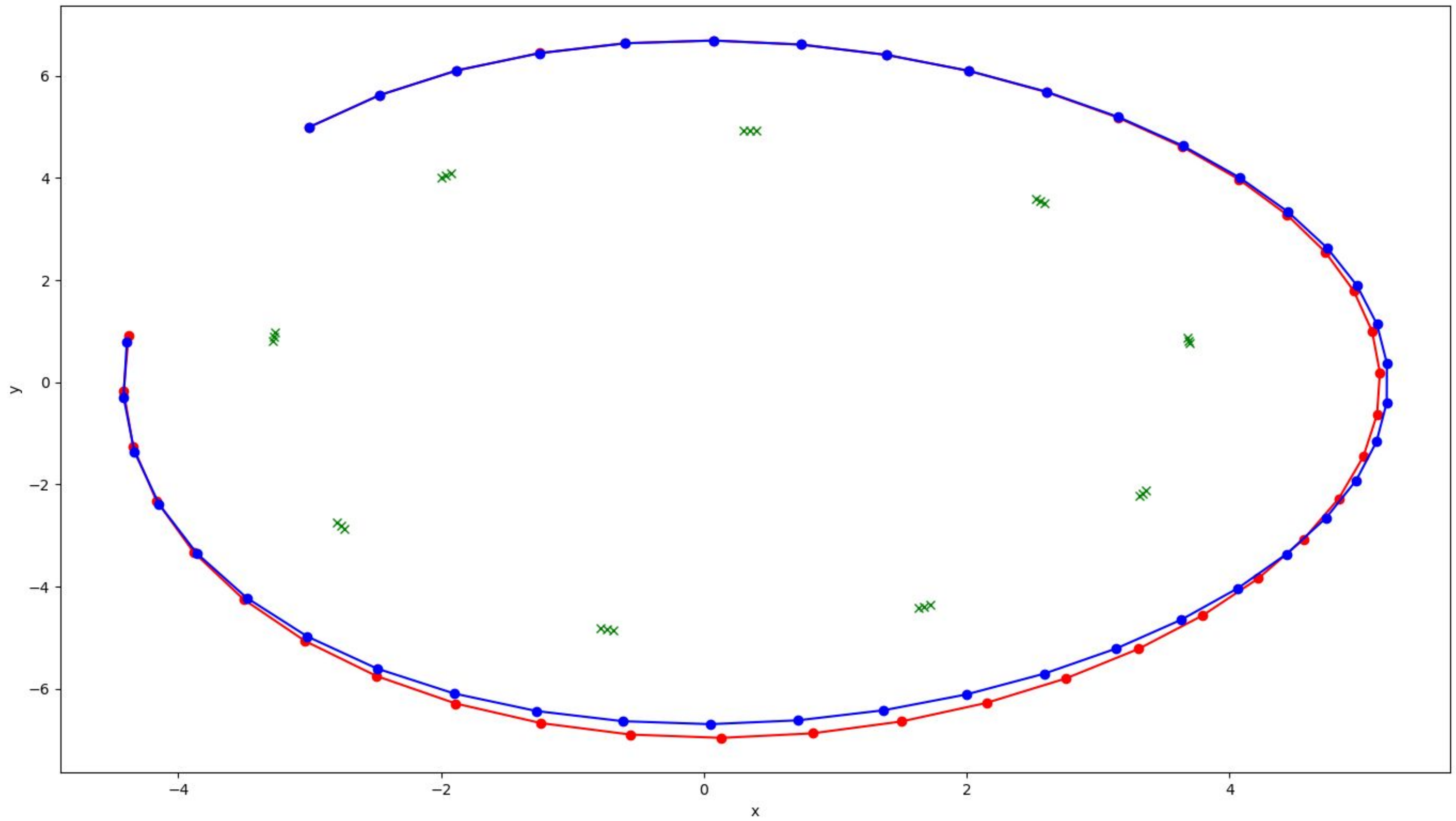


Figure 2. Training data used for ODEs reconstruction (greed dots), system trajectory generated by PNN for another initial condition (red line), real system trajectory (blue line)

Beam dynamical system

General view of the equation of motion in the form of Taylor mapping:

$$X = \sum_{i=0}^{k} \mathbb{R}^{1i}(t) X_0^{[i]}$$

$$X_0 = \begin{pmatrix} X_1 \\ Y_1 \\ Px_1 \\ Py_1 \\ \dots \end{pmatrix}$$

Beam dynamical system

General view of the equation of motion in the form of Taylor mapping:

$$X = \sum_{i=0}^{k} \mathbb{R}^{1i}(t) X_0^{[i]}$$

$$X_0 = \begin{pmatrix} X_1 \\ Y_1 \\ Px_1 \\ Py_1 \\ \dots \end{pmatrix}$$

$$\mathbf{X} = \sum_{i=0}^{k} \mathbb{R}^{1i}(t) \mathbf{X}_0^{[i]}$$

$$\mathbf{X}_0 = \begin{pmatrix} X_1 & X_2 & \dots & X_n \\ Y_1 & XY_2 & \dots & Y_n \\ Px_1 & Px_2 & \dots & Px_n \\ Py_1 & Py_2 & \dots & Py_n \\ \dots & \dots & \dots & \dots \end{pmatrix}$$

St Petersburg University
www.spbu.ru

Beam dynamical system

Linear case

$$\mathbf{X} = \mathbb{R}^{[11]}\mathbf{X}_0^{[1]}$$

Non-Linear case:

$$\mathbf{X} = \mathbb{R}^{[11]}\mathbf{X}_0^{[1]} + \mathbb{R}^{[12]}\mathbf{X}_0^{[2]} + \dots$$

Kronecker power of the 3 order:

$$\mathbf{X}^{[3]} = \begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{X}^{[3]} \\ \mathbf{X}^{[2]} \bigotimes \mathbf{Y} \\ \mathbf{X} \bigotimes \mathbf{Y}^{[2]} \\ \mathbf{Y}^{[3]} \end{pmatrix}.$$

Beam dynamical system

For example, for quadrupole:

$$\mathbb{P}^{quadrupole} = \begin{pmatrix} \mathbb{P}^{11} & \mathbb{O} & \mathbb{P}^{13} & \dots \\ \mathbb{O} & \mathbb{P}^{22} & \mathbb{P}^{23} & \dots \\ \vdots & \vdots & \vdots & \ddots \\ \mathbb{O} & \mathbb{O} & \mathbb{P}^{jk} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

$$\mathbb{A}(1,1) = h\frac{13(\cos(l\sqrt{|h|}) - \cos(3l\sqrt{|h|}) - 36l\sqrt{|h|}\sin(l\sqrt{|h|})}{192}$$

$$\mathbb{A}(1,2) = \sqrt{|h|}\frac{-5\sin(l\sqrt{|h|}) - 13\sin(3l\sqrt{|h|}) + 12l\sqrt{|h|}cos(l\sqrt{|h|})}{64}$$

Conclusion

- Standard machine learning algorithms can hardly provide physical interpretation of the considered dynamical process

- Proposes an architecture and learning algorithm for polynomial neural networks based on Taylor maps for dynamical system learning with small datasets

- The Taylor mapping technique provides transformation of the dynamic system represented in the form of ODEs to the initial weights of the polynomial neural network

- Work is in progress.

Thank you for your attention!

Neural Networks in Modeling Beam Dynamics using Taylor Mapping*

Nataliia Kulabukhova, Anna Golovkina, Vladimir Kozynchenko

*work is supported by Saint Petersburg University for the research grant ID: 75206008.